

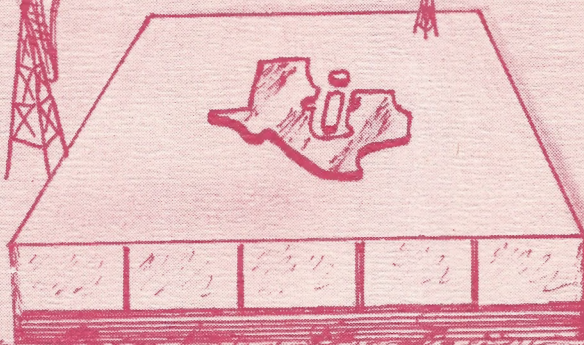
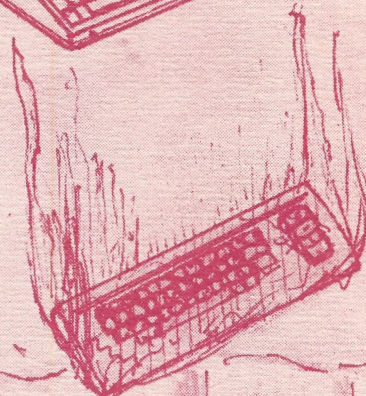
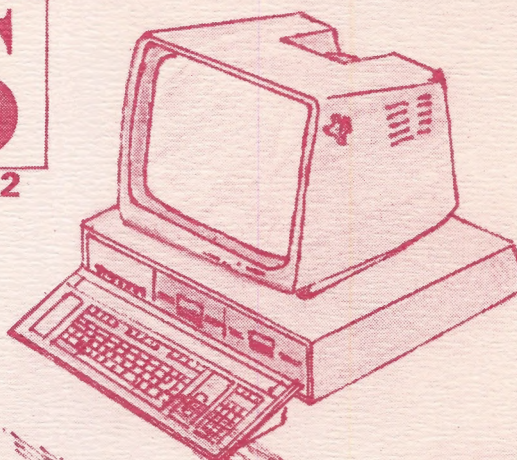
# SYDNEY NEWS

SEPTEMBER '85

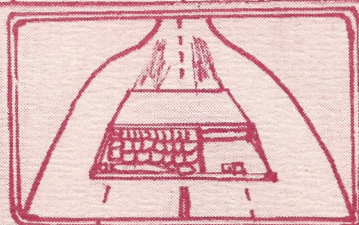
\$2

**HUG**  
PO BOX 149  
PENNANT HILLS  
2120, N.S.W.

REGISTERED BY AUSTRALIA POST  
PUBLICATION No. NBH5933



# DIGEST



\*BUMPER ISSUE



## EDITORIAL

Welcome to another issue of the Sydney News Digest. You may have noticed that this issue is a bumper size with some very interesting new features.

### FIRST INSERT

In the envelope with this publication, you will find two other sheets, one being a cover which I have designed for you to place into a Spring ring binder. The Picture was created by the very talented Younger Set member ... master Sam Mudie who was a second place winner in Jenny's Younger Set UNDER 18's FRONT COVER COMPETITION.

This special cover is a once only card with a limited print run. So place it, along with the centre PROGRAM LISTINGS into a binder to start a special book of TI-99/4(A) programs to be typed in.

I have designed this publication so that all future PROGRAM LISTINGS for you to type in, can be removed with ease, and placed into your very own Binder, which, intime, will build up to make a large book of listings that you will be proud to own.

I hope that you like the idea.

### SECOND INSERT

This double sided page has been created by Fred Morris, our new Publication Librarian. This is part of a new MEMBERSHIP DRIVE for TI.S.H.U.G., and one which you will reap the benefit of. I hope that you will make the effort to join up some of the friends who you know to be TI Users.

### OUR NEXT MEETING

Our 2nd BIG AUCTION day. Saturday the 7th of September (2pm) will be the next big club activity.

As usual it will be conducted at the St.John's Church Hall in Victoria St, Darlinghurst (near St.Vincent's Hospital). It will be another Auction Afternoon.

The afternoon will be split into two (2) sections:

1. SWAP-A-THON.
2. CLUB AUCTION.

With such an even, there have to be guidelines which we ask that you strictly follow... and they will be explained further in this article.

THE SWAP-A-THON will be conducted during the first half hour of this afternoon. This will be an opportunity for you to exchange modules with others at that meeting.

CLUB AUCTION. On the back page of this Sydney News Digest, you will find an Auction List. You are asked to either cut it out or photocopy it. But you must bring an Auction List along with you.

- (1) Any person not supplying and Auction List, and obtaining a LOT NUMBER from the Auctioneer, will not have his/her goods Auctioned on the day.
- (2) Only MODULES and PERIPHERALS will be allowed for Auction. Disk/Cassette software and TI-99/(A)computers will not be permitted for sale, swap or auction.
- (3) Each and every item you bring into the hall MUST be tagged or labelled with your NAME on them. A record will be kept of all people buying and selling.

(4) ALL ITEMS SOLD:via CASH or Club BANKCARD ONLY. Regarding the Club Bankcard: This service will be available for your convenience with a service charge of 10% to cover handling costs...and AS-IS basis. The Club takes no responsibility as to the condition of goods sold.

(5) ALL SALES will be finalised at the fall of the hammer. No bid shall be retracted, and in the case of dispute, the Auctioneers decision shall be final.

(6) Each person will be allowed to Auction 5 items. If there is time at the end, then anything else you have, will be put up also...and an Auction List will then be provided.

(7) The seller does not have to sell, unless he/she is happy with the price.

(8) CHARGES: Those bringing MODULES - \$2.00 (any quantity). PERIPHERALS - \$2.00 EACH ITEM.

So there you have it. It should be a fun and productive afternoon for all, and a great time to thin-out or stock-up goodies for your TI-99/4(A) Computer. As they say on TV...BRING YOUR MONEY WITH YOU ... but don't forget your Auction List.

### IN THIS ISSUE >

Many of our Survey forms have now returned to us with interesting results. It was interesting to read that half the membership want the SND to cater more for BEGINNERS and the other half of the membership want the SND to cater more for the EXPERIENCED programmer. Well, you can't please every one all the time. This particular issue will no doubt please those who are interested in ASSEMBLER, MINI MEMORY, and other related information. Other issues will cater for the others as time goes by.

If you can contribute in other areas/articles & tutorials, then you are most welcome to do so. But in the past 4 1/2 years that this large club has been in operation, the response to that plea has been nothing less than pathetic. We have a lot of talent in this group, but many just won't take the time to get off your butts and do something with that talent, so, more overseas content is provided. I should take this opportunity once again to thank the Channel 99 Canadian User Group for the wealth of information they have provided us and which we in turn pass onto you. These articles are written by W.A.JOHNSON, DON COOK, A.CHARRON, and MIKE TOWERS.

A special thanks go to the following TI.S.H.U.G members for their contribution to this bumper issue...KEIR WELLS, FRED MORRIS, ROSS & SAM MUDIE, TERRY PHILIPS, JOHN ROBINSON, ALAN SWALES, BRIAN GRAHAM, D.N. HARRIS, JENNY and the many Younger Set members who constantly show-up the adults with their continued efforts.

There are now those who have contacted Fred Morris, to offer their assistance in retyping both articles and programs from our overseas and national publications and who form the team on the PUBLICAIONS COMMITTEE...Thanks gang! You help my job flow just that much faster, and this issue is the start and proof of some of that effort.

### OCTOBER MEETING

Now this should be a good one... it is hoped that we will be taking over the WOODSTOCK MANSION for a very special FULL-DAY TUTORIAL WORKSHOP. More details about that event next month as it is co-ordinated. WOODSTOCK MANSION is situated in Church Street BURWOOD and will be a refreshing change to our existing meeting place. We do need lecturers in EXTENDED BASIC, ASSMBLER, FORTH, and other areas. If you can help with this, please contact either John Robinson or Fred Morris on the numbers provided on side two of the special PROGRAMMERS BOOK cover, AS SOON AS POSSIBLE.

Well, thats about it from me, now to update the ELECTRONIC MAGAZINE (TEXPAC BBS) with goodies for those with modems.

*Shane*

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$  Tips on M U L T I P L A N  $
$                               $
$   by Brian Graham            $
$ with extracts from the 99'er $
$ International TI Magazine.   $
$                               $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

MULTIPLAN FOR THE TI.99/4A -- AN INTRODUCTION TO SPREADSHEETS.  
PATRICIA SWIFT, TECHNICAL EDITOR, 99'er HOME COMPUTER Magazine. A REPRINT OF SOME OF THE ARTICLES FROM 99'er HOME COMPUTER AS  
EDITED BY BRIAN GRAHAM.

Multiplan is meant for all. It can predict changes in budgets for the household based on changes in the rate of inflation. It can be used to compare holiday costings. And it can run your own business. These things and more. It can be used as a data base and when combined with TI WRITER is just marvellous. Heard this all before. Is the salesman who said all this still in business. Chances are that if he had and used MULTIPLAN he probably is. Anyway MULTIPLAN is still around in many systems and is not really understood by all as to its potentiality. I use it to run the accounts of a Club I belong to and also the Accounts of a Charity. It is indeed a timesaving tool when you apply all of its facilities.

I must confess that I was a little confused when I sat down and waded through the excellent manual that accompanied the module and disk. Oh yes. YOU NEED A DISK DRIVE TO RUN MULTIPLAN TOGETHER WITH EXPANDED MEMORY.

Microsoft's MULTIPLAN is a "second generation" spreadsheet package, is easy to learn and use (with a little time and patience plus a few tips on shortcuts). It allows you to sort the information on the spreadsheet and even allows linking of worksheets so that information can be shared among them.

STILL INTERESTED. Then this article or series of articles is for you.

The minimum micro-computer for business would have 48K of main memory, two disk drives, and invariably a printer. Because it is a sophisticated program, MULTIPLAN needs 48K of memory but will work on one drive although two would help and be convenient especially when you want to invoke the HELP !!! functions which are user friendly. I survive with minimum trauma with only one drive as I rarely need the HELP capabilities of the package. Unless you want a copy of the sheet you don't even need a printer.

Once you have created your model or template, feed in the information and MULTIPLAN does the rest. Well almost. Hopefully this article will help you get the most so that the rest can be done by MULTIPLAN. With the ability to link worksheets if you update data in one and another model is reliant on that data in the first model MULTIPLAN will automatically update that other model. MULTIPLAN is quite flexible. The largest worksheet it can handle contains 63 columns by 255 rows. Needless to say the size of the worksheet is governed by the main memory of your computer configuration but the ability to link is handy. MULTIPLAN constantly displays the percentage capacity remaining in any on worksheet so there is little danger of suddenly running out of room.

Output from MULTIPLAN can be saved to disk for editing via TI WRITER so you can prepare tables using MULTIPLAN with all its calculation abilities and then use the

table in a report or an article with accompanying text. Talking of calculations MULTIPLAN automatically recalculates any cells dependent on the changed cell BEFORE YOU ARE ALLOWED TO ENTER FURTHER DATA. This can be tedious especially if your favourite long last aunt has just dropped by for that once a year cuppa. DON'T DESPAIR. ALL IS NOT LOST. This automatic recalculation can be turned off and then you don't have to wait for the recalculation after every change. More on this latter. Still intrigued. Then you must either want to find out how to easily use this powerful tool or you have nothing better to do with your valuable time.

MULTIPLAN is not just for business. How many kids, or should I say young people, out there use a calculator for your Maths homework. Why not put MULTIPLAN to work for you. Once you have set up your mathematical model you only need to feed in the data. With a calculator how many times have you forgotten to feed in a component used in your calculations. How annoying to have to do it all over again if you cannot see your mistakes in your formulae. MULTIPLAN to the rescue as it does not forget any part of the formulae unless like me you make just a few mistakes in entering it in the first place. With MULTIPLAN you have the ability to edit formulae without re-entering the entire thing from the very start. What a time saver. More time to catch up on your other activities.

READ the manual that comes with MULTIPLAN to fully understand how it all happens. What follows now is some of the strategies for using some of MULTIPLAN's theories and the overall package.  
WHAT MULTIPLAN CANNOT DO.

In spite of its editing capabilities, MULTIPLAN is not a word processor, but this is not to say that it could not be used to create and edit the type of documents usually produced by using the TI WRITE word processing package for example. It just would not be very convenient for the user because of the row and column layout, but it is not totally impossible. We will leave it to your ingenuity to attempt same. Nor does MULTIPLAN shine in situations where quantities of information must be entered into the computer and then selectively distributed for processing as in an invoice situation. However, against with a little Aussie know-how all is not impossible. Like anything a little thought at the planning stage of your model and the use of linked worksheets could allow this type of function to be attained from MULTIPLAN. A specialised accounting package may be easier to use and there are many on the market for the TI 44/9A; just write to one of the warehouses listed in the SND from time to time. MULTIPLAN is also limited in the amount of information each worksheet can hold, but this can be overcome by linking them together. details on how to do this can be found in the manual accompanying the programme.

Really, it is a question of what absolutely could not be done on MULTIPLAN but rather the construction of a good model from the very start and how much time it would take to get MULTIPLAN to produce the desired result. Once you understand and learn how to use and adapt the programme you will probably be able to devise a way to use MULTIPLAN for everything you do.

MULTIPLAN TEMPLATES.

Remember as you are limited to only 63 columns and 255 rows you may wish to sit down before hand and draw a rough sketch of your worksheet. When you finally sit down to use the programme you may find that the column width is not to your liking. You can alter the default of 8 characters to start or you can do that after all your information and data have been entered. It makes no difference. You can go as wide as 32 characters and MULTIPLAN makes the job easy by letting you type in the full description but only displaying as many characters as will fit in the current width of the column on display. When you widen the column later, the longer description will automatically appear. This even happens when you are windowing and entering numbers and figures. Just because it cannot be seen entirely on the screen does not mean it is not all there. Formatting

on MULTIPLAN is just like formatting on any wordprocessing package. It is something that can be done after all your data has been entered.

## MEANINGFUL INTERACTION.

As you merrily plug away number crunching, you may wish to alter the layout as you go. NO PROBLEM. MULTIPLAN is highly interactive and user friendly, is convenient and has a give and take relationship with you. In addition to anticipating your wishes with the default for column width and format, MULTIPLAN always allows you the user to override the suggestions it makes to you. You are after all THE BOSS.

You will notice that as you get more information on a worksheet the longer it takes for MULTIPLAN to accept the next input from you. This is because of the automatic re-calc facility built into the programme. Each time you fill in a cell, MULTIPLAN checks all other cells to see if their contents should be changed to reflect the data you just entered. The bigger the worksheet the longer it will take. ALAS, all is not lost. You can turn OFF the re-calc facility while you enter the data and polish your worksheet, and then turn it on again when you want MULTIPLAN to perform calculations. The OPTIONS command lets you do this. Additionally, if the re-calc is turned off and you forget to turn it back on then if you attempt to save the data to disk then the programme will automatically recalculate the sheet before it is saved. Dummies of the world have been saved, but only if saving to disk. If you want to print the sheet without saving it then you will have to turn the re-calc back on. Therefore it makes sense to save to disk first and let MULTIPLAN take over our laziness. If all else fails don't forget the HELP !!! capabilities of MULTIPLAN. This can be accessed either through the OPTIONS command or by entering a question mark at any time except in the middle of text ( the ALPHA command ).

ABOVE ALL, TEST YOUR WORKSHEET TO MAKE SURE IT DOES ALL YOU WANT IT TO DO. HOW YOU DO THIS I LEAVE IN YOUR CAPABLE HANDS.

Once the worksheet is finished and stored on disk the real power of MULTIPLAN can be used. This is its ability to evaluate similar situations containing different information so that a template once saved can be used over and over again given the same model requirements. The nice thing about using these electronic models ( in addition to the time saving factor ) is that you can experiment without the fear of loosing your original model.

## MULTIPLAN EDITING COMMANDS.

To err is human, to foul up is like me. To the rescue comes the editing features which help us out of our jam. No need to retype the whole cell, whip into edit. The most obvious area is if we use a Command mnemonic letter when we really wanted something quite different. The CANCEL key [ CTRL = ] will cancel unwanted commands but to make sure this does not happen often it is best to select the Alpha command before entering any text beginning with a Command letter. To delete the previous character typed use the BACKSPACE key [ FNCT 9 ], or the DELETE FORWARD [ FNCT 0 ] destructive edit key. The BACKSPACE edit key deletes to the left of the cursor; the DELETE FORWARD deletes to the right of and under the cursor. This comes in handy in editing any long formulae one used in the calculation. The format code called continuous can also be used to edit long pieces of text. This is done by moving the cursor to the start of the text and selecting the FORMAT command by typing "F", and then the option, CELLS. MULTIPLAN will assume that the cursor position is at the start of the cell to be formatted. To indicate that the format should extend to the right type ":", and move the cursor to the right via the arrow key to the end of the section to be edited. Tab [ CTRL 2 ] over to the alignment selections and type "G" for General. The Tab to the code selections and type "C" for continuous. Press [ENTER] and MULTIPLAN takes in the new continuous format. Easy wasn't it ? Try again if not successful first time. Now all you have to do is select Alpha Command, type in the

replacement information and follow it with the [ENTER] key. SIMPLE !!!

You can also edit another way. You can move the cursor back to the bad spot and by using the WORD BACK [ FCTN 5 ] and CHAR BACK [ FCTN 4 ] edit keys. Move the cursor using WORD BACK until you reach the error and then use CHAR BACK to put the cursor over the wrong letter and delete by pressing BACKSPACE [ FCTN 9 ] once. Insert letters the same way only use CHAR FORWARD [ FCTN 4 ] instead of CHAR BACK and type in the missing letter. Try it out. VOILA !!! The error is fixed without massive retyping. Press [ENTER] to tell MULTIPLAN you're finished. Life was not meant to be easy but why not shorten methods and let MULTIPLAN do the work for you. In summary, the nondestructive commands are CHAR BACK, WORD BACK, CHAR FORWARD and WORD FORWARD. The destructive commands are BACKSPACE and DELETE FORWARD. The DELETE FORWARD key removes whatever is highlighted by the edit cursor while the BACKSPACE key removes just one character at a time to the left of the edit cursor. BEWARE of the "?". Under edit command this cannot be entered anywhere in the text as it is the signal for HELP and if you do this is what you will need and you will find the once friendly programme is no longer friendly, and may be down right antagonistic. Editing does require a little practice and a dash more patience.

Apologies to Patricia Smith for some of the comments which are my own and not hers.

## HINTS TO MAKE MULTIPLAN WORK FASTER.

\* Obtain from the TISHUG shop the disk entitled " TIMP " and this will accelerate the use of the MULTIPLAN programme by allowing faster cursor movements in scrolling the screen and a few other goodies not found on the original disk that comes with the package.

\* After loading your model or template use the OPTIONS to turn off the automatic re-calculation. This is done moving into command mode and typing " 0 " and then typing " N ". Using this only the cell being worked on is changed. During the period that re-calculation has been turned off you can still at any time do calculations simply by pressing the RECALC key [ FCTN 8 ]. Remember also that even if you don't press the RECALC key before saving your worksheet to disk, MULTIPLAN will do this automatically and believe me this aspect can come in mighty handy for us forgetful people.

\* Avoid extensive use of forward references as they are slower to recalculate. For example, a reference to cell R10C10 from R5C5 is slower than a reference to R5C5 from R10C10. Therefore try moving the cursor to the bottom right hand corner of the worksheet [ CTRL 1 ] before you hit RECALC key [ FCTN 8 ]. try it and see.

\* Use compacted formulae instead of adding all cells one at a time add them by range. This means using the SUM provisions found in the manual. It makes the formula tidier and easier to see if need be. You may only have to check one line of formula for errors instead of three lines. For example, it is much tidier to use SUM[RC[-12]:RC[-1]] THAN using the old way of : R[-12]C+R[-11]C+R[-10]C+R[-9]C+R[-8]C+R[-7]C+R[-6]C+R[-5]C+R[-4]C+R[-3]C +R[-2]C+ R[-1]C WHERE ALL WE WANT TO DO IS ADD A COLUMN OF NUMBERS. You will also find that this overcomes the problem of trying to build a long formula which will not be accepted by the programme and you end up using about three formula to get the end result.

\* Lock all cells containing formulae so that you will not accidentally overwrite one that has taken time to figure out thus saving headaches and frustration when you come to use a model and find only garbage is the result. To do this place the cursor on the cell containing the formula to be protected, Press " L " for lock, then Press " F " for formula. Lastly enter " Y " to confirm when prompted. ALL FORMULAE WILL NOW BE



PROTECTED FROM OVERWRITING until you change this protection.

\* Do not fill unneeded cells with zeros as these will use up valuable memory and be taken into account when recalculating. It is easier and faster to leave them blank or to BLANK them out if accidentally filled with zeros unless it is absolutely necessary to fill with zeros.

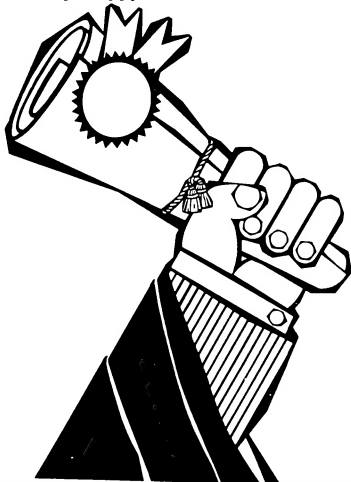
\* If a formula is used more than once in any worksheet, use the COPY command. This saves time for us lazies.

\* CHAPTER 9 OF THE MANUAL MAKES INTERESTING READING and starts at page 121 especially at pages 139, 143, 146, 150 and 160.

\* APPENDIX 6 STARTING PAGE 225 IS ALSO HELPFUL.

Hoping that these edited articles and other comments are of benefit to those of us who use MULTIPLAN, I hope also that those of you who might be considering a spreadsheet programme for home or business use that this has whetted your appetite for more. Don't forget the LIVERPOOL REGIONAL MEETING on FRIDAY 11th. OCTOBER, 1985 will concentrate on MULTIPLAN. Contact CYRIL BOHLSSEN on PHONE (02) 639-5847 FOR ALL DETAILS OR THE MICROSOFT MULTIPLAN USERS GROUP, P.O. BOX 95, FORESTVILLE. NSW. Perhaps if enough of use want to we could form a special interest group within TISHUG if one is not already operating.

Bye for now and keep happy with TISHUG and MULTIPLAN.



## PREGNANCY PROBLEMS LINKED TO TERMINALS

A report released in Japan says more than a third of pregnant women working on VDUs have problems during pregnancy or delivery.

The report was published by Japan's General Council of Trade Unions, which has 4.5 million members.

Thirteen-thousand workers, of whom some 4500 were women, were surveyed. Two-hundred and fifty were reported as having become pregnant or given birth after working at the Terminals.

The average time spent in front of a screen was two years and seven months.

Of the 250 women, 91 were said to have experienced problems ranging from miscarriages and premature births to stillbirths.

The figure was significantly higher for pregnant women in the group who worked for six hours a day or more at the VDUs.

Two-thirds of these had problems. The report identified tension as a factor, as well as eye strain and fatigue.

## TI99-OPOLY RECALLED

A note from the program author, Ross Mudie.  
ref:TI99-OPOLY.REVISION1 31/7/85.

Following comment from Russel Welham and a note in the Glebe regional report, (SND August 1985), it is agreed that the program contains rule interpretation errors with regard to:

- (a) Lack of follow on actions as a result of jumps caused by 5 of the 16 Chance cards.
- (b) The method of handling doubles in the dice routine.

Ross is currently working on these two problems, but is unaware of any interpretation errors in the case of any of the Community Chest cards.

Any person with criticism of the program is encouraged to contact Ross as soon as possible on work phone (02) 663 0171, (7.30am-2.30pm), so that these may be considered.

It is proposed to make the modified program available at no charge, (except postage one way), to persons who return both the original TI99-OPOLY disk and booklet.

Owners of TI99-OPOLY, serial numbers 1 to 23 (except 11 & 15) should contact Ross Mudie on the above phone number or at the address in the front of the TI99-OPOLY booklet, to permit a list to be compiled for the change over when the modified program is proven and the booklet is revised.

Further to Peter Schubert's suggestion to attempt to make the game loadable from tape into a computer with memory expansion. Unfortunately, no way is seen to achieve this at present, due to the program length (approximately 23K bytes of extended basic program) and the apparent lack of support of CALL LOAD for assembly routines from cassette.

SELL: TI RS232 Interface PHP1220,  
suit TI Peripheral Expansion Unit  
\$140. Ross Mudie ph (02) 663 0171,  
7.30am-2.30pm weekdays.



Recommendations contained in the report include: pregnant women should not work at VDUs until it can be shown that there are no potential problems in pregnancy or labour; no worker should be allowed to work at VDUs for more than four hours a day and workers should be given a 10-minute rest break every hour.

The Japanese statistics fly in the face of advice handed out in some other countries, including the UK.

In Britain it has been said that around 15 per cent of pregnancies end in miscarriages in the general population, the use of VDUs notwithstanding.

A recent article in New Scientist, however, said no studies had been comprehensive enough to actually clear VDUs, although examinations of the equipment failed to detect enough radiation to damage health.

Eyestrain and backache were the worst hazards of the technology, according to the New Scientist report.

NOW HERE IS SOME HELPFUL ADVICE FROM PETER HANCOCK OF T.I.C.H.U.G.....

"All unmarried females should stand at least 3 metres from terminals just in case!!!!

WOULD YOU LIKE A TERMINAL FOR A FATHER?"



## I LOVE COMMODORE

by W.M. Johnson

O'k the game's up I have to come forward and tell the truth. If you have one of those ignominious little twerps following you around who says; "Oh you've got a T.I.", then snickers.

DO NOT blush!

DO NOT apologize!

DO NOT hold your poor little 16 bit 16K High Tech miracle behind your back and smile weakly.

HOLD you head up high straighten you back and say, "Yup, I'm not a sucker for crass commercialism." or "Oh, dear you must be one of those ingrates that accidentally bought a, what was it?"

I got rid of my mentor by forcing him to see the DIMness of his ways.

"Look at it," he said "it's not even a real keyboard".

"True" I replied. "But it not only works, it keeps working."

"It's only 16K 'giggle, giggle'"

"True" I replied, "but it's 16 real K of 16 bit with 22 bit stacks, not a phantom 64K with imaginary ROM and 8 stingy bits."

I got tired of the quips. "Hay Commodore Turkey, I admit all, why don't you bring your wonderful machine over and show me for instance how that marvelous sound generator works."

"SURE", his pride enveloping him like an aura.

He soon returned, I had great difficulty controlling the giggles. I watched fidgety as he gently placed his eggshell like box of tricks on the table. I couldn't help thinking how it looked like a sun bleached flattened wiener with keys. He hooked it up to my old T.V. just as if he knew what he was doing.

"I tell you what", I said with a sardonic smile "Now that we've got both on the same table why don't we both program a tune?" I had hit the right spot, his face paled.

"Well, er well, O'k", he stuttered.

"How about this one, it's short. 'Ghost Riders'?"

There was no answer, he nodded affirmatively. I could see his breath begin to quicken. I swear I could see his chest pounding as his heart went into high gear.

Suddenly feeling calm and undaunted, I opened the music book with a flourish, then like Boris Brott (Conductor), spectacles at the end of my nose I opened my manual to page III-7. Work commenced.

Internally I giggled continuously for the next 40 minutes, THEN "I've finished!" I exclaimed.

"What?" gasped my wide eyed companion. "You couldn't be."

With confidence, poise, and grace I pressed "R" then "U" then "N". the room fell into a breathless hush <ENTER>.

Well, it wasn't exactly the Hamilton Philharmonic, and there were a couple of sharps where there should have been naturals. BUT I DID IT, and in only 40 minutes

"There," I said when the music fell silent.

"Wasn't very good," he remarked.

"Well let's hear your's.!"

The Commodore user thought for a minute, "I've got a great Tree with Christmas music on tape."

"So what," I growled I want to hear GHOST RIDERS." reluctantly he pressed the appropriate keys. "None of them fell off this time."

I hadn't laughed so much since the last Richard Pryor movie. After all that time all he got was: "PING, PING, RING, PING." Trying desperately to control my mirth I said "is that it?"

"Yes", he said meekly. The real problem is that the 64 is, in computer terms "prehistoric technology". Modern machines are very clever, with great amounts of knowledge built in, 'ROM'. The T.I. has over 70K of ROM with another 30K in your extended basic. The Commodore has only 64K, end of message.

The sound system of the 64 is a classic example of under design, slipshod construction and a total lack of forethought.

Let me give you an example.

T.I. Ansi/soft basic:--

10 FOR I=0 TO 30

20 CALL SOUND(-50,440,I)

30 NEXT I

A very simple program that makes a tone that tapers off into the distance.

Commodore Basic:--

10 SID=54272

20 FOR I=0 TO 28 : POKE SID+1,0 : NEXT

30 POKE SID+24,15

40 POKE SID+1,20

50 POKE SID+5,0+0

60 POKE SID+6,15,+9

70 POKE SID+4,1+16

80 POKE SID+4,16

You may ask who is this poor Sid that is being poked to death. In Commodore SID stands for Sound Interface Device. It may be difficult to believe but the two examples above produce almost the same sound. Of course this demonstrates why your average Commodore USER is just that a USER not a programmer.

Most programs in Commodore Basic are in fact in pseudo machine language, but let's stay with sound I'll knock all the rest another time.

"You've got no control over your pitch, attack or decay on your T.I.". That's a good and common line used by Commodore users. Believe me it's not true. The control of the voices on the T.I. is not as good as the Commodore, BUT it does not take 3 years in university to get something that sounds almost as good.

It takes a special person to program music into the 64, he or she must have a good understanding of SOUND, MUSIC, and PROGRAMMING in machine language. On your old T.I. all you need is sheet music and page III-7 of you users guide.

So why do I love Commodore?

Easy, I run a business selling and repairing computers. You can't make a living repairing the Maytag of the computer world, ie. Texas Instruments. On the other hand we have Commodore ????

But don't go off half cocked, the Commodore 64 is an excellent computer, and if programmed properly will perform extremely well. It's sound capabilities are slightly better than that of the T.I. and that's all.

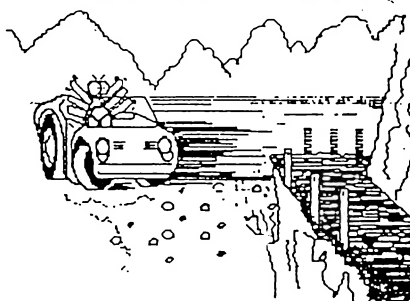
For a few extra bucks you can get some real sound. I put a reverb (home built), a moderate 30 watt stereo amplifier, and a pair of decent speakers on my system.

Even the games are mind boggling when you wind up the power and blast those aliens. I can't imagine why there's a lot of houses for sale in our neighbourhood

The opinions expressed in this article are those of the writer and not Ch 99 or the publishers of Club-Line

**END**

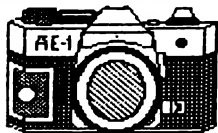
### GRAPHX DEMO \*\*\*\*\*



SEE REVIEW ELSE WHERE IN THIS NEWSLETTER



### A full-service graphics package



MADE ON GRAPHX IN LESS  
THAN 30 MINUTES FROM  
STORED A LITTLE CLIPBOARD  
AND A LITTLE WORK!

## MICROpendium

(C. BOBBITT)



The following article has been reproduced from MICROPENDIUM, an American magazine entirely devoted to the TI 99/4A. MICROPENDIUM is a publication well worth subscribing to, being one of America's most widely read TI magazines.

By CHRIS BOBBITT

For better or worse, there aren't very many truly useful programs available for the 99/4A.

This is really a shame considering that before most people bought the machine, they had some particular purpose in mind—a productive task that could be made simpler with the use of a computer.

Most, even those with the best of intentions, have ended up using the computer solely for entertainment. This is usually because most people can't locate software that enables them to do something on the computer that they can't do elsewhere much easier.

Fortunately, there are a few exceptions. TI-Writer, Multiplan and a number of third-party word processors, mailing list programs, terminal emulators and database programs are excellent tools that make computers productive. These are tools that make your machine more than just a sophisticated Colecovision by turning it into, for a time anyway, a true "productivity tool."

In the area of computer graphics, the category of "productivity tools" is nonexistent. Most other machines have programs that are useful for creating business graphics, incorporating graphics into text or performing even more esoteric but useful tasks. The TI99/4A, in part because its graphics are harder to manipulate, hasn't inspired similar applications programs, at least until now. I'd like to introduce GRAPHX, which is truly, as the subtitle states, "The Ultimate Graphics Software."

This Australian import is different from the myriad of other drawing programs and peripherals available for the 4A. It can be used for something other than creating pretty pictures, even though it can do that too. This program can be easily used to make near typeset-quality text in an infinite number of fonts, create logos and advertising material (even whole ads) and create detailed scientific and business graphs and charts.

**Performance:** GRAPHX comes on a single disk, and can be loaded through either the Editor/Assembler or the Extended BASIC cartridge. It takes about as long to load this program as the others. (In other words, count on sitting around for about a minute or so

until the program is ready to go.)

After loading, the program draws a fantastic title screen, consisting of a detailed drawing of a 99/4A console and monitor, and an ever-changing monitor "screen." If you could sit and watch, you would see 16 different pictures, from a furry koala bear to a fine drawing of the Space Shuttle, displayed on the drawing's monitor. The program continues to display these pictures until a key is pressed.

After the user presses any key, the screen clears, a star-shaped cursor appears and a little help message is displayed at the top of the screen, indicating to the user that he or she should press any of the program's function keys. The 10 most often used functions of the program are accessible by pressing any of the number keys. The remaining functions are available through a series of pull-down menus that can be accessed by pressing the "=" key. After the menu key is pressed, the menu scrolls down (while carefully preserving your picture underneath it), and an option of the menu is highlighted in a different color (green). To choose an option, highlight the appropriate choice by moving the joystick up or down, and then by pressing the "fire" button. The program uses no other keys; one never has to touch the FCTN, CTRL or SHIFT keys while this program is running. Most input is by joystick, except for those 11 keys at the top. Except for Super Sketch, this program probably has the least complicated of any of the drawing programs.

GRAPHX has a lot of features that set it apart from the other graphics programs. However, the whole genre carries some similarities. Like all graphics programs, it permits the user to use a joystick or other input device (keys, etc.) to draw on the screen in any of the 15 available colors. Like most graphics programs, it can be used to change the color of objects, change the screen colors, fill shapes and erase objects. Like some, it can be used to draw lines and other mundane shapes, and it can save and load pictures from diskette. And like a few, it contains a built-in, Epson-compatible screen print option. However, this particular program contains some features never seen in a program like it for the TI99/4A.

First of all, the program has a powerful "zoom" option. When the "6" key is pressed, a little window-shaped object appears on the screen. The window can be moved to any place on the screen simply by pulling the joystick in the appropriate direction. When the "fire" button is pressed, the portion of the drawing in the window instantly becomes four times larger, and fills the television or monitor screen. This option is excellent for fine, detailed work where a single extra pixel may throw off an entire picture. In the zoom mode, the cursor can be set to any of the five available speeds, and one can draw or erase pixels with the joystick at will. To leave this mode, simply press "6" again, and the screen becomes normal, with all changes implemented. The speed of this routine is spectacular.

GRAPHX also has one of the best line and circle drawing functions ever. Lines and circles are created with a "rubber-banding" method. In other words, a line or circle appears on the screen after "8" or "9", respectively, is pressed, and the shape can be stretched, flattened, made smaller or made larger by moving the joystick in the appropriate direction.

To choose, and thus draw, the desired shape, simply press the "fire" button. An added feature; circles, once their shape has been decided upon, can be moved anywhere around the screen before being placed permanently. This is useful for creating complex figures. The shapes and sizes of the circles and lines are for all purposes infinite.

This program has two features that every graphics program should have but none else do, the ability to move or copy any portion of the screen in either black and white or color. It's very simple to operate. After the "9" key is pressed, a short menu appears. After any of the four available move or copy options is selected, a 16x16 pixel box is displayed on the screen. The box can be made larger or smaller simply by moving the joystick. The user has a choice of four box sizes, 16x16 pixels, 32x32 pixels, 48x48 pixels or 64x64 pixels.

After a size is chosen, the box chosen becomes mobile. It can be moved anywhere on the screen simply by pushing the joystick in the desired direction. After it's on the desired portion of the picture, simply press the "fire" button. A black shadow of the portion of the picture to be copied or moved appears in place of the box. Depending upon the option chosen at

the short menu, one of two things happens. If in the copy mode, when the shadow is moved with the joystick, the original remains. Using the joystick and the "fire" button (used to drop the picture), one can place an infinite number of copies of the shadow all over the screen before returning to the short menu. If in the move mode, after the shadow is moved a blank area remains where the portion of the picture was removed. To drop the shadow in its new location, again press the "fire" button. These two functions are extremely utilitarian and have a distinctly useful purpose.

The features already listed would be enough for almost any graphics program on the market today, and with the exceptions of the move and copy options, there are graphics programs that duplicate each and every one to some degree. However, this program wouldn't be a true productivity tool if it lacked the last, but most important set of features: the "clipboard" options.

The clipboard, simply put, is an alternate area of memory where many small pictures, even special alphabets and clip art, can be stored until needed, separate from the picture seen on the screen. This file of pictures can be saved and loaded separately from the screen. Thus, the user can build up a library of thousands of fonts and pictures that can be copied anywhere onto other pictures at any time.

All the clipboard features are accessible through a sub-menu of the main menu. Every conceivable option was included. The user can save or load clipboard files from disk, view clipboards without disturbing the picture, put copies of the pictures in the clipboard file any place on the screen, put pieces of the picture on the screen into the clipboard file and delete certain pictures in the file or the entire file.



The program is provided with a number of example clipboard files, including computer, gothic and normal letter fonts, and two animation sequences (more on that later). Letters or pictures are placed on the screen simply by selecting the desired picture by leafing through the clipboard, and then by moving the selected object to the desired destination. Shades of MacPaint!

With the clipboard the user can even experiment with computer anima-

## SHADES OF MACPAINT

tion—not the sprite variety but instead that used to make movies; in other words, one frame at a time. After drawing each frame in the screen, simply remove them to the clipboard in the desired order (they are placed sequentially in the file from the first removed to the last), and use the "see" option of the clipboard menu to look at them. You can leaf through the frames at a constant rate of about three or four per second by simply holding the joystick pushed forward. The clipboard will hold about 64 16x16 pixel pictures, 16 32x32 pixel pictures, seven 48x48 pixel drawings or four 64x64 pixel ones. Of course, you can save pictures of various sizes in the same file.

GRAPHX has a number of miscellaneous functions worth mentioning—a typewriter mode where you can move a cursor any place on the screen and type uppercase letters anywhere (great for labeling), a print option that allows screen dumps in two sizes and two densities, the five-speed cursor, an option to turn off the ever-present help lines temporarily, an erase option that puts grey blocks in a checkerboard pattern all over the screen and much more.

I've been using this program for two months now, and except for one minor bug, every function seems to work perfectly. The bug? In zoom mode, the eight pixels in the upper right corner are always transparent, even though they are really another color in the regular mode. It's not a disabling error, just mildly irritating. I have only one other minor complaint which really isn't very fair to the program. I wish it had a catalog disk option (even though none of the others do). This is mainly because clipboard files tend to proliferate like rabbits. A number of times I've had to quit the program in order to catalog my data disks to find the one containing the file I need.

**Ease of Use:** GRAPHX, unlike the grandfather of graphics programs, Draw-A-Bit (and unfortunately some of its progeny), is very simple to use. The only keys used by the program are the 11 in the top row, the various letter keys and ENTER for such things as entering filenames and in the typewriter mode. All other functions are available through those fascinating pull-down menus and the joystick.

The documentation of the program recommends that the user make a function strip for the top row—and even gives a sample diagram of what it should look like. With this function strip and a good joystick, and by

following the multitude of "help" prompts, this program is exceedingly simple to use. Despite the program's simplicity, you can create some really complex graphics scenes the equal of the best made with other graphics peripherals or programs. With the addition of the numerous functions and menu options described, this program is a first class drawing program, almost as versatile and nearly as sophisticated

as the \$1,000 Auto/CAD program for the TI-Pro. It's also a heck of a lot cheaper and so much easier to use.

**Documentation:** The manual provided with this program is almost unnecessary. Despite this, it's probably one of the best manuals for any program on this computer or any other. The manual provides instructions and exercises for learning how to use the program, and it's very useful as a detailed reference source on the operation of all its options.

For the first-time user, the manual contains numerous diagrams and printouts from the program itself to explain options. It includes very understandable descriptions, and even lists possible uses of options, yet remains completely comprehensible. For the technical user interested in adapting pictures or clipboard files into his own assembly programs, the documentation includes a considerable amount of pertinent information. The program disk even includes the assembly language source code for a program to read and display saved picture files. The manual is professionally done.

**Value:** As mentioned, this program is the first "productivity tool" for graphics for the 99/4A. If you own a business, work in an office, or need to prepare reports for school, this program will meet and perhaps exceed your graphics requirements. It can create beautiful charts and graphs, fine title displays suitable for photocopying, quality text in any conceivable font, diagrams and pictures suitable for business and school as well as scientific work. If you are a professional artist, it may not meet your total graphics needs, but a suitable system for an artist may cost thousands of dollars.

The program is also exceptionally enjoyable to use, letting even those of us who can't even draw a straight line create exceptional quality pictures in 15 brilliant colors. For a mere \$50 this program exceeds the specifications of MousePaint (\$135) on the PCjr, and many other programs. If you have a professional or casual interest in graphics, this program is for you. Next to TI-Writer, I personally use no other program

### Report Card

**Performance** .....A  
**Ease of Use** .....A  
**Documentation** .....A  
**Value** .....A+  
**Final Grade** .....A

### Cost: \$50

**Manufacturer:** R.L. and C.P. Davis,  
P.O. Box C568, Clarence Str., Sidney,  
Australia, 2000

**Requirements:** Console, monitor or television, expansion memory, Extended BASIC or Editor/Assembler, disk system, Epson Mx-80 or compatible printer recommended



HOLD ALL THE  
ACES  
RENEW YOUR  
MEMBERSHIP  
NOW!

### TI99/4A INTERACTIVE ADVENTURES

Newest, intriguing text-only excitement! Discover "Temple of Terror" and "Cave-In." Ransack an ancient and deadly Greek temple, avoiding traps, cobras and the feared Brotherhood. Or explore a definitely hazardous cavern to find the lost

"Borealis Bullion" while avoiding outlaws and hazards enough to test your deductive abilities to the limit. To order, send \$10 each (BASIC-cassette), \$12 (XBASIC cassette—Harder!), or \$15 (XBASIC Disk—Most Complex!). For both XBASIC adventures send \$20 (cass.) or \$25 (disk). We pay postage. FOXWARE, 167 Watt St. 60-7, Battle Mountain, NV 89820.

### THE FAITHFUL FEW

When the meeting's called to order  
And you look around the room,  
You're sure to see some faces  
That from out of the shadows loom,  
They are always at the meeting  
And they'll stay until it's through—  
The ones that I would mention  
Are always the faithful few.

They fill the many offices  
And are always on the spot,  
No matter what the weather,  
Though it may be awful hot;  
It may be dark and rainy,  
But they are tried and true—  
The ones that you can rely on  
Are always the faithful few.

There are lots of worthy members  
Who will come when in the mood,  
When everything's convenient  
They can do a little good.  
They're a factor in the meeting  
And are necessary, too—  
But the ones who never fail us  
Are always the faithful few.

If it were not for these faithful,  
Whose shoulders at the wheel  
Keep the institution moving  
Without a halt or reel,  
What would be the fate of meetings  
Where we claim so much to do?  
They surely would be failures  
If we lacked the faithful few.

THANKS GO TO THE ATTIC (Adelaide TI User Group) FOR THIS LITTLE PIECE. TIMELY TOO?  
- ED.





## LINKING TO ASSEMBLY FROM EXTENDED BASIC.

By Ross Mudie of TISHUG.

TESTING the DEF TABLE from extended basic.

When one or more assembly language programs are loaded into the low area of RAM (>2000 - >3FFF) under control of extended basic, the computer sets up a table of the names of the entry points into the assembly language programs. This is known as the DEF table.

It is possible to utilise CALL PEEK from extended basic to look at the DEF table and check if a particular assembly language program has already been loaded. The advantage of this is when a different extended basic program is loaded, any previously loaded assembly program is unaffected, providing that the computer remains in extended basic and unnecessary reloading of the assembly program can be avoided.

The DEF table contains entries of 8 bytes each in length, the first 6 bytes are the program entry point name (which appeared in the DEF statement in the source file) and the next 2 bytes point to the memory address of the entry point into the program. To calculate this address (which is not necessary for the present discussion but is an interesting aside) multiply the first address byte by 256 and add the value of the second. All these values are decimal numbers.

The first entry in the DEF table is at (hex) >3FF8 through to >3FFF, ie, decimal 16376 to 16383. If there was only one assembly program used and only one entry point in memory it would be at 16376. If the program entry point name was START, then the result of:

CALL PEEK(16376,A,B,C,D,E,F):: PRINT A;B;C;D;E;F  
would be 83 84 65 82 84 32. Compare the ascii codes for the letters of START to the results. The reason for the 32 (space) is that the computer allows 6 bytes for the program name and shorter names are padded out with ascii 32's. The following program example is suitable if only one assembly language program is used with an extended basic program.

```
130 CALL PEEK(16376,A,B,C,D,E,F):: IF A=83 AND B=84
AND C=65 AND D=82 AND E=84 AND F=32 THEN 160
140 CALL INIT
150 CALL LOAD("DSK1.ASSYPROG")
160 ! normal program continues here
```

## MULTIPLE ASSEMBLY LANGUAGE PROGRAMS.

Once more than one assembly language routine is used where extended basic is either too slow or the facility doesn't exist, it is likely that your program will need to search the DEF table for matching program entry point names. The first entry in the table is at 16376 and the last entry in the table is pointed to by the two bytes at (decimal) 8196, (>2004). The following program example checks if CALL INIT has been performed and if there are any entries in the DEF table by looking at the values in memory location 8196. CALL INIT is performed only if it has not been previously performed. If there are assembly programs in memory then the names are PEEKed and converted to strings in the T\$ array. Comparisons are then carried out for the required programs and the disk names for those not in memory are placed in the N\$ array. The required programs are then loaded from disk.

```
110 CALL CLEAR
120 CALL PEEK(8196,X,Y)!Pointer to last entry in DEF
table
130 IF X<63 OR X>64 THEN CALL INIT ! Init is only
called if not previously executed
140 IF X=64 AND Y=0 THEN 230 ! It is pointless testing
DEF table if it is empty
150 FOR S=16376 TO X*256+Y STEP -8 ! This works thru
DEF table from first entry to last entry
160 CALL PEEK(S,T(1),T(2),T(3),T(4),T(5),T(6))! Six
ascii characters for name of program entry point
170 FOR RT=1 TO 6 ! Reads the ascii values from T
array & makes entry point names in T$ array
180 IF T(RT)=32 THEN 210
190 T$(C)=T$(C)&CHR$(T(RT))
200 NEXT RT
210 C=C+1 ! C is an index counter for T$ array usage
220 NEXT S
230 FOR TE=0 TO C :: IF T$(TE)="START" THEN 250 ! If
START is found then it doesn't need to be loaded again
240 NEXT TE :: N$(1)="START" ! The program identified
on disk as "ASSYPROG" has an entry point "START"
250 FOR TE=0 TO C :: IF T$(TE)="K" THEN 270 ! If K is
found then it doesn't need to be loaded again
260 NEXT TE :: N$(2)="KEY" ! The program identified on
disk as "KEY" has an entry point "K"
270 FOR S=1 TO 2 ! Substitute "TO" number for the
actual quantity of programs used
280 IF N$(S)="" THEN 290 ELSE CALL LOAD("DSK1."&N$(S))
290 NEXT S
```

## WHAT HAPPENS WHEN GOING TO & FROM OTHER CARTRIDGES?

When using an assembly routine from extended basic then changing to other modules, such as editor assembler or disk manager, (using a cartridge expander) I found that some cartridges did not affect my assembly language utility whilst another affected part of the utility in low memory.

The Disk Manager does not appear to affect the utility in the low memory area and on return to extended basic it is not necessary to reload the utility.

Using the EDIT1 program from Editor Assembler, part of the low memory area becomes over written but the DEF table at the top of the low memory area may remain intact. No doubt there are much more elegant methods to overcome this problem than I will present but here are some ideas which may start others thinking.

The following changes refer to the above program which tests for multiple entry point names in the DEF table.

Line 130 becomes:

```
130 IF X<63 OR X>64 THEN CALL INIT ELSE 140
```

Then add line 135 which destroys the first byte of each entry in the DEF table after INIT has been called:

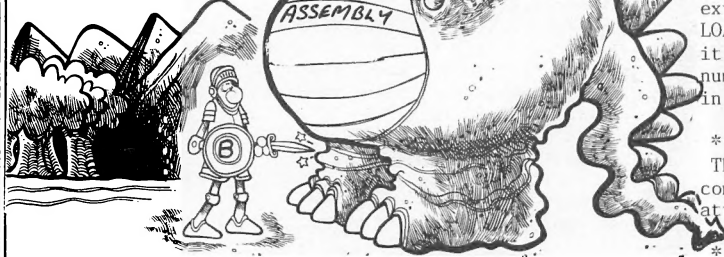
```
135 FOR S=16376 TO 16296 STEP -8 :: CALL LOAD(S,0) ::
NEXT S
```

It is also advisable to limit the number of names tested, I chose to limit to 10 as follows:

```
210 C=C+1 :: IF C=10 THEN 230
```

This then allows full reloading of the utilities by the extended basic program and permits operation in the extension memory without having to turn the power off the expansion system to remove unwanted programs from the extension memory.

STAND-ALONE MULTI-MODEM/RS232 UNIT is now going into production (by Peter Schulbert). This unit which was shown at the TEXPAC BBS PARTY is now fully functional and can be used with T.E.II module to access this BBS and all similar services. The RS232 port can be connected to a printer and can be accessed thru the T.E.II output function(CTRL 2). I am now working on the software to access VIATEL and this will be made available as soon as possible. Price of the unit is \$280 at present and I am now taking orders. Phone me (Peter) on (02)358 5602 AH. If you know of someone who would like to COMMUNICATE I ask you to spread the word and make our BBS grow and prosper. My regards to all. From PETESAKE.



## TWO NEW UTILITIES. by Ross Mudie of TISHUG.

An assembly language program has been written to provide two convenient to use, fast utilities which can be linked from an extended basic program on a TI99/4a home computer which has memory expansion and a disk drive attached. That program is available through the club Librarian. See him about details of distribution

These utilities offer the following features:

### (a) Single press key routine "K"

- \* Similar to CALL KEY.
- \* Flashing, unique cursor at any screen location.
- \* Cursor pattern specified in the extended basic program which permits the cursor to be easily changed, even while the program is running.
- \* A return variable is provided for any of the 5 TI99/4a key unit scan modes.
- \* The character corresponding to the key pressed replaces the cursor at the same location.
- \* The utility waits for the key pressed to be released before returning to the extended basic program.
- \* The utility can be simply instructed to delete the screen character when the key is released if required.
- \* The keyboard is scanned approximately 900 times per second whilst the cursor is flashing.

### (b) Clear Screen Routine.

- \* Allows clearing of any part of a screen, (not a whole screen), more quickly than can be achieved in extended basic, without disturbing the rest of the screen.

Both utilities incorporate default values which prevent missing or invalid parameters from terminating a running program. Once loaded into the low area of the extension memory under extended basic program control, these utilities do not require reloading whilst the computer remains in extended basic, even if other extended basic programs are loaded and run.

The command syntax is compatible with the existing command structure.

The extended basic program KEYUTILITY is used as a convenient method of transporting the object file and provides a means of testing the routines prior to inclusion in other programs. Help files named HELPKKEY55 or HELPKKEY70 (both have the same text) have been produced, which fully explain the utilities.

.....

## KEY UTILITY 'K'

- \* Format.  
CALL LINK("K"[,key-unit,row,screen-column][,return-variable])
- \* Description.  
The CALL LINK("K") utility is similar to CALL KEY in

that it responds to a single key press from the key board and assigns the code of the key pressed to the Return-Variable. The utility allows values to be passed from the extended basic program to position a unique cursor at any position on the screen. The row is any value from 1 to 24 and the column any value from 1 to 32. Non integer values will be rounded down to the integer value. This utility must be loaded into extension memory by including CALL INIT and CALL LOAD("DSKx.KEY") in the extended basic program prior to its use. (The 'x' refers to the appropriate disk drive number on which the "KEY" object code utility resides in DISPlay FIXed 80 format).

### \* Hardware Limitation.

This utility is only suitable for TI99/4a home computers with memory expansion and a disk drive attached.

### \* Application.

When the utility is required in an extended basic program, include the statement:

```
CALL LINK("K",KU,R,C,RV).
```

The variable KU is the key unit and may be any value from 0 to 5. The variable 'R' is the screen row number and may be any value from 1 to 24.

The variable 'C' is the screen column number and may be any value from 1 to 32. The Key Unit, Row and Column variable names may be any valid numeric variable name or number values, e.g., CALL LINK("K",5,12,25,RV) will place the cursor in row 12, column 25. The value of KU affects the code returned from a key press, a 5 will return both upper and lower case alphabetical and numbers etc. If the KU value is outside the range of 0 to 5 then the codes returned, will by default, be as if KU=0. If the Row and/or Column values are outside the valid range or are omitted then the cursor will default to row 24, column 3. The defaults for row and column can apply individually or together. The use of CALL LINK("K") without arguments will use the default cursor position (R=24 & C=3) and key unit (KU=0), in addition, no return variable will be available. The Return Variable 'RV' may be any valid numeric variable name, however it must appear in the fourth position in the argument list. The code returned from the key press will correspond to the key board mapping applicable at the time that the utility was entered, refer to key unit value summary below. The unique, redefinable cursor will flash at the specified location until a key is pressed. If the key pressed corresponds to a screen printable character then this character will be displayed in place of the cursor and the program will wait for the key to be released before returning to the extended basic program. This 'wait' feature prevents the program from skipping through following instructions if the key is not immediately released.

The cursor pattern must be defined early in the extended basic program, this is to allow full flexibility for the user. A suggested cursor pattern may be defined using CALL CHAR(127,"000000424242E"), however refer to the User's Reference Guide II-76 to 79 to design a unique pattern for the cursor. If the cursor is not defined then no cursor will be displayed, but if there is a character on the screen at the cursor row and column it will be deleted.

### \* Method of avoiding unnecessary reloading.

To reduce unnecessary reloading of the KEY utility every time the extended basic program is run, by including CALL PEEK(16376,A,B):: IF A=75 AND B=32 THEN [LINE NUMBER following CALL LOAD("DSK1.KEY")] will test that the utility is in memory and skip the reloading; see the example in the sample program. If another extended basic program is loaded without turning off the memory expansion or going to Editor assembler etc, the KEY utility may still be in memory and not require to be reloaded. The PEEK at 16376 checks the first two characters of the first entry in the DEF table for the program name "K" (ascii 75) followed by a space (32). If it is likely that more than one entry will be present in the DEF table, (due to other assembly language programs being in memory), then other entries in the table may need to be tested. The entries are 8 bytes long each, the second entry starts at 16368 going up to 16375. The first 6 characters are the program name and the next two numbers point to the entry point memory location.



## Prompt Beep.

An input prompt beep may be provided in the extended basic program by including: CALL SOUND(150,1400,4):: CALL SOUND(10,1400,30) immediately before the CALL LINK("K") statement. The second CALL SOUND is required to ensure that the beep is turned off before entering the assembly language utility, otherwise the beep tone will not turn off until the return to extended basic.

## \* Sample Program.

```
100 CALL CLEAR
110 CALL PEEK(16376,A,B):: IF A=75 AND B=32 THEN 160
120 PRINT "LOADING KEY UTILITY...."
130 CALL INIT
140 CALL LOAD("DSK1.KEY")
150 CALL CLEAR
160 CALL CHAR(127,"0000004242427E")
170 INPUT "Row?":R
180 INPUT "Column?":C
190 CALL KEY(O,K,S):: IF S=1 THEN 190
200 CALL SOUND(150,1400,4):: CALL SOUND(10,1400,30)
210 CALL LINK("K",5,R,C,RV)
220 PRINT "Return Variable is";RV:
230 PRINT "Enter next..."
240 GOTO 170
```

## \* Cursor Address outside Basic Text area.

The basic text area uses screen columns 3 to 30 and these are defined by some of the text print commands as columns 1 to 28. By using a value of 2 or 31 in the Column variable (C) of the CALL LINK("K",KU,R,C,RV) argument list, the special cursor may be displayed and the result of the key press printed without disturbance to the screen full of text. If it is desired to remove the character left by the key press, then specify the value of KU as negative, e.g. CALL LINK("K",-3,R,C,RV). As the utility returns to extended basic after the pressed key is released, the character resulting from the key press will be replaced with a space character.

## \* Key Unit Value Summary.

The value of the numeric expression passed to the KEY utility in KU affects the code returned from a key press. Full details are included in the User Reference Guide II-87 to 89, however a summary is provided below.

KU=0 Key unit should be as previously defined in a CALL LINK("K") or CALL KEY, if the previous definition was 3, 4 or 5. Use of KU=0 is not recommended.

KU=1 The left side of the key board and joy stick 1 fire button will be scanned, codes 1 to 19 are returned.

KU=2 The right side of the key board and joy stick 2 fire button will be scanned, codes 1 to 19 are returned.

KU=3 Numeric will be normal, lower case alphabetical will be returned in upper case and function codes 1 to 15 will be active.

KU=4 Numeric will be normal, upper and lower case will both be active, function codes 129 to 143 and control character codes 1 to 31 will be available.

KU=5 Numeric will be normal, upper and lower case will both be active, function codes 1 to 15 and control character codes 128 to 159,187 will be available.

If the above values of KU are specified as negative then the absolute value of KU will be taken and there will be no change to the keyboard scan. The negative value will cause the character displayed on the screen, as a result of the key press, to be deleted when the key is released.

## \* Changing the Cursor Character Byte.

It is quite feasible to change the character byte in the object code if you wish to use another character number for the cursor. The byte is in characters 20 and 21 of the first line of the object code. It is the DF in 00258KEYmudieA0000BDF. The byte value represents the hex equivalent of the ascii character code plus an offset of hex 60. The standard cursor (ascii 30) is hex 1E. Hex 1E + hex 60 = 7E, thus if the DF were replaced with 7E the normal cursor would be always given and line 160 of the sample program would be unnecessary. It is open to the programmer's choice but the use of a unique cursor for the single key press entry is recommended for user recognition of the single key press operation at that point. If the byte is changed in the first line then the checksum tag for

that line must be changed from 7 to 8. The checksum tag is the 59th character in the first line. The file must be resaved in DISPlay FIXed 80 format using the file name KEY, e.g., DSK1.KEY.

## \* Scan Speed.

This routine can be emulated in basic or extended basic using CALL KEY, a counter, CALL HCHAR and an IF THEN statement in a loop, however due to the time taken by basic to traverse the loop a quick key press may be missed.

The CALL LINK("K") utility scans the keyboard approximately 900 times per second when the cursor is flashing and does not miss a quick key press.

## KEY UTILITY 'CLS'

## \* Format.

CALL LINK("CLS",[Row Start],[Row Finish],[Col Start],[Col Finish])

\* Description. The CLS utility allows the clearing of a portion of a screen more rapidly than using HCHAR or DISPLAY AT in a FOR-NEXT loop in extended basic.

'CLS' is included in the object code module KEY with the 'K' utility. The module loads with 'K', refer to the description of KEY for loading instructions.

## \* Application.

In an extended basic program when it is desired to clear only a portion of the screen include the statement:

CALL LINK("CLS",RS,RF,CS,CF).

RS is the Row Start; RF the Row Finish; CS the Column Start; CF the Column Finish. The names of these variables may be any valid variable name or numeric values may be used. Any non-integer value will be reduced to the integer value.

In the event of a value being passed to the utility out of the valid range, a default value will be substituted for the invalid value. The following lists the valid value range and the default values:

Variable	Function	range	default value
RS	Row Start	1 to 24	1
RF	Row Finish	1 to 24	24
CS	Column Start	1 to 32	1
CF	Column Finish	1 to 32	32

The arguments in CALL LINK("CLS") must appear in the order given, however if one or more arguments are omitted then the arguments will be deleted from the right hand end of the list and default values will be assumed.

If the finish value is smaller than the start value then the finish value will become the same as the start value, resulting in the clearance of only one row or column for that pair of variables, e.g. CALL LINK("CLS",11,4,15,9) will produce the same results as CALL LINK("CLS",11,11,15,15); the character at row 11 column 15 will be replaced by a space.

## \* Other Examples.

CALL LINK("CLS") will clear the whole screen but CALL CLEAR is more efficient.

CALL LINK("CLS",4) clears all columns from row 4 to row 24 inclusive.

CALL LINK("CLS",6,12) clears all columns from row 6 to row 12 inclusive.

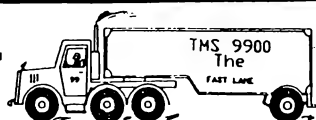
CALL LINK("CLS",18,22,15) clears columns 15 to 32 inclusive from row 18 to row 22 inclusive.

CALL LINK("CLS",10,14,7,17) clears a window from column 7 to 17 inclusive on rows 10 to 14 inclusive.



## The Fast Lane

EDITED BY  
Iain Johnson



### DISK SECTORS CONTINUED

by  
Don Cook

Below is a catalogue listing of a disk and sectors used on the disk to illustrate how the information is stored on the disk.

DISKNAME=68DSKMG

360 SECTORS, 69 AVAILABLE

FILENAME	1.INFO	SIZE	TYPE	2.USE	3.DATE
*CATALOGUE		3	PROG.		MAY19/85
CASSTO	<CASSTS	12	DF 80	OBJECT	MAY07/85
CASSTS	(DISKS)	6	DV 80	SOURCE	MAY07/85
CATS1	cCATS3	21	DV 80	SOURCE	MAY12/85
CATS3	>CO	27	DV 80	SOURCE	MAY19/85
CO	<CATS3	24	DF 80	OBJECT	MAY19/85
DIM	DSK.MGR.	33	PROG.	ASSEM.	MAY19/85
DIMHELP	HelpFile	19	DV 80	DATA	MAY09/85
DISKS	cDISKS1	20	DV 80	SOURCE	MAY08/85
DISKS1	>DO	28	DV 80	SOURCE	MAY19/85
DMS	<DMS	22	DF 80	OBJECT	MAY19/85
DMS	>DMS	16	DV 80	SOURCE	MAY19/85
DO	<DISKS1	23	DF 80	OBJECT	MAY19/85
INITS	cDMS	7	DV 80	SOURCE	MAY04/85
MAINSUB	cDMS	6	DV 80	SOURCE	MAY04/85

The table below lists the sectors used (in HEX notation) for each file.

68DSKMG	DISK SECTORS USED
*CATALOGUE	011,031-032
CASSTO	00F,106-10F,111
CASSTS	00B,102-105,110
CATS1	004,04D-05D,091,131-132
CATS3	009,033-04C
CO	00C,092-0A7,101
DIM	00A,0A8-0B0,0B2-0C7,12F
DIMHELP	003,0C8,0F0-OFF,130
DISKS	002,022-030,112-114,116
DISKS1	005,05E-06D,08A-08B,0B1,0C9-0D0
DMS	00E,0DC-OEF,117
DMS	006,06E-07B,133
DO	00D,119-12E
INITS	007,07C-081
MAINSUB	008,08C-090

The table below lists the hexadecimal values in sector 0. Refer to the March newsletter to determine the purpose of each byte value.

68DSKMG	Sector 0
HEX values	ASCII values
0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF
00 3638 4453 4B4D 4752 2020 0168 0944 534B 68DSKMG	*HDSK
10 2028 0101 0000 0000 0000 0000 0000	(*****
20 0000 0000 0000 0000 0000 0000 0000	*****
30 0000 0000 0000 0000 0000 0000 0000	*****
40 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
50 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
60 0000 0000 0000 FFFF FFFF FFFF FFFF	*****
70 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
80 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
90 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
A0 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
B0 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
C0 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
D0 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
E0 FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****
FC FFFF FFFF FFFF FFFF FFFF FFFF FFFF	*****

### 68DSKMG Sector 1

0 1 2 3 4 5 6 7 8 9 A B C D E F  
00 0011 000F 000B 0004 0009 000C 000A 0003  
10 0002 0005 000E 0006 000D 0007 0008 0000  
\*\*\*\*\* Bytes >20 to >FF are all >00 \*\*\*\*\*

This sector provides the information for a file named DISKS. Note that the word at >0A has been utilized to code the file use and date and bytes >14 to >1B have been used for file information.

### 68DSKMG Sector 2

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF  
00 4449 534B 5320 2020 2020 A885 8003 0013 DISKS \*\*\*\*\*  
10 F650 1300 7E44 4953 4B53 3120 22E0 0012 \*P\*\*cDISKS1 \*\*\*\*\*  
20 1101 1621 0100 0000 0000 0000 0000 0000 \*\*\*\*\*  
\*\*\*\*\* Bytes >30 to >FF are all >00 \*\*\*\*\*

Bytes >1C to >24 are used to point to the sectors used by the file. The values >22, >E0 and >00 make two word values >022 and >00E to point to sector >022 as the first file sector through to sector >030 (>022 + >00E). The next 3 bytes >12, >11 and >01 make the words >112 and >011, indicating that the next file sectors are at >112 to >114 (>112 + >011 - >00E ->001). Note that the ->001 calculation was missed in the March newsletter. The next 3 bytes make up the word values >116 and >012 to point to sector >116 as the last file sector used. **END**

### TI(D)BITS

by A. Charron

CALL LOAD(-31700,4) .....LAST SOUND NEVER STOPS

CALL LOAD(-31740,A,B).....A B = VALUES FOR DIFFERENT SOUNDS

CALL LOAD(-31748,1) .....NORMAL CURSOR SPEED AND WARNING SOUND HIGHER NUMBER CURSOR BLINKS FASTER

CALL LOAD(-31803,35) .....GOES BYE LIKE QUIT

CALL LOAD(-31804,X) .....X=ANY NUMBER...BE PREPARED TO TURN YOUR MACHINE OFF SEVERAL TIMES...ALSO MAKE SURE NO DISK IN

CALL LOAD(-31806,16) .....DISABLE QUIT KEY BIT 3 ON

CALL LOAD(-31806,32) .....KILLS AUTO SOUND BIT 2 ON

CALL LOAD(-31806,64) .....KILLS SPRITES BIT 1 ON

CALL LOAD(-31806,128).....KILLS ALL ABOVE BIT 0 ON

CALL LOAD(-31806,0) .....RESTORES ABOVE ALL BITS OFF

CALL LOAD(-31806,48) .....SOUND/QUIT OFF BITS 2 & 3 ON

CALL LOAD(-31806,80) .....SPRITES/QUIT OFF BITS 1 & 3 ON

CALL LOAD(-31806,96) .....SPRITES/SOUND OFF BITS 1 & 2 ON

CALL LOAD(-31888,63,255)..WILL SHUT DOWN ALL DISK DRIVES

CALL LOAD(-31888,55,215)..TURNS THEM BACK ON THE ABOVE NOT FULLY TESTED MAKE A COPY OF PRGM



## The Communicators



Welcome to another COMMUNICATORS section.  
Below is a listing of programs that are available to down-load on the TEXPAC ELECTRONIC MAGAZINE (BBS).

New features are being added to this unique service of our club and one which we feel you should know about has made our BBS even faster to use.

Shane Ferrett has given us the FAST DISPLAY FEATURE. This is how it works...

Up to a short time ago, every screen page stoppen to ask you if you wanted to continue. Now the sysystem will continue to scroll up with no stopping until it gets to the end of each bulletin. But you can stop that scrolling when ever you wish, with <CTRL> S and simply press any key to continue, or press 'E' to EXIT either the Main Menu or News Menu.

When you want to print a screen, first stop the text from scrolling with Control S then press Control 2 to out-put that information to either disk or printer.

As you may be aware, the TE#2 will store up to 3 screens of information then sound a BLEEP to let you know that you have 8 seconds to stop the scrolling and save that information before it writes over itself.

With the introduction of of VIATEL, many of you have purchased dual speed modems, making it possible for you to access 1200/75 baud BBS's. Our TEXPAC Electronic Magazine can also run at 1200/75 baud. We plan to set aside a day or part of the week-end to those with the capabilities to access it at that speed. Read all about it on our system. If you want to access our BBS at the higher speed, then get in touch with SHANE on the BBS so that we can have some indicatin as to whether its worth the effort.

And now...here are the programs you can obtain from this TEXPAC ELECTRONIC MAGAZINE...

NAME	SIZE	TYPE	COMMENT
ALIGATOR	11	Bas	Game
BOA-ALLEY	17	Bas	Game
DRIVING	20	Bas	Game
GRAPHMATCH	13	Bas	Game
JAWBREAKER	21	XBas	Game
LAS-VEGAS	33	XBas	Game
RISING-SUN	17	XBas	Music
S-DESIGN	28	XBas	Screen Utility

Next month(SEPTEMBER)we have the following for you...

NAME	SIZE	TYPE	COMMENT
AZTEC	23	Bas	Printing Utility
BOWL6	28	XBas	Game
COLOR-VIS	25	Bas	Screen Graphics
DISKLABEL	9	XBas	Print Disk Catalog
LOAD	5	XBas	Loading Utility
PUMKIN	14	XBas	Screen Graphics
SAUCERS/B	19	Bas	Game

If there is a particular type of program you would like to down-load... Leave Mail with your request to Username:GRACE

```
CALL LOAD(-31878,X) .....X= HIGHEST # OF
                        SPRITES YOU ARE GOING TO
                        USE
```

```
CALL LOAD(-31931,16) .....TURN TRACE ON
```

```
CALL LOAD(-31931,0) .....UNPROTECT PROGRAM
CALL LOAD(-31931,16)OR 128..PROTECT PROGRAM
```

```
CALL LOAD(-31962,255).....WHILE IN XBASIC
                        WILL LOOK FOR A PROGAM
                        CALLED LOAD
```

```
CALL LOAD(-32572,1) .....GIVES MUSHIE
                        KEYBOARD
```

```
CALL LOAD(-32572,128).....DISABLES KEYBOARD
```

```
CALL LOAD(-32700,0) .....CLEARS SCREEN FOR
                        A SPLIT SECOND
```

```
CALL PEEK(2,A,B) .....THEN CALL
                        LOAD(-31804,A,B)***BRINGS
                        UP TITLE SCREEN
```

```
CALL PEEK(-28972,A) .....IF A=96 SPEECH
                        SYNT. ATTACHED..IF A=0
                        THEN NO SPEECH SYNT.
                        ATTACHED
```

```
CALL PEEK(-31808,A,B).....RANDOM NUMBER
                        FROM 0 TO 255
```

```
CALL PEEK(-31879,A) .....VDP
                        INTERRUPT....COUNTS 1 TO
                        100 IN 4.25 SEC
```

```
CALL PEEK(-31880,A) .....A= RANDOM # FROM 0
                        TO 99
```

```
CALL PEEK(-31888,A,B).....PRINT A6+B =
                        HIGHEST VDP ADDRESS
```

```
CALL PEEK(-31888,A) .....A= AMOUNT OF
                        MEMORY USED
```

```
CALL PEEK(-31938,A,B).....PRINT A6+B =
                        START OF BASIC VARIABLES
                        NOT REALLY
                        SURE???????????
```

```
CALL PEEK(-31950,A,B).....PRINT A6+B = END
                        OF LINE # ..START OF
                        TUAL ACTUAL TOKENIZED
                        BASIC PROGRAM
```

```
CALL PEEK(-31952,A,B).....PRINT A6+B =
                        LOWEST FREE MEMORY FOR A
                        BASIC PROGRAM..START OF
                        LINE # TABLE
```

```
CALL PEEK(-31974,A,B).....PRINT A6+B-1776
                        IS THE SAME AS SIZE IN
                        XBASIC
```

```
CALL POKEV(-32272,0,"",-30945,0)..THIS WILL
                        DO 40 COLUMN FOR A SPLIT
                        SECOND
```

```
CALL POKEV(-32280,0) AND POKEV(-32766,0)
                        FOR OTHER VDP DISPLAY
                        COLOR BLOCK AND BIT MAP
```

```
CALL POKEV(-32768,0).....RESETS CHANGES
                        MADE IN -32766
```



## BAR GRAPH

by Don Cook

The program listed below illustrates a method of directly checking either joystick and displaying a horizontal bar graph on the screen . Pressing either joystick left or right will increase or decrease the bar on the screen . Pressing a joystick up or down changes the speed of reaction . The FIRE button causes a QUIT .

```

DEF BARS
REF VSW,VMBW
TITLE TEXT '0 20 40 60 80 100'
MWS BSS 58
BARS LWPI MWS
*****
*Define bar graph characters in VRAM*
*****
LI R3,9      9 Characters to define
LI R0,>000    VRAM location for ASCII 128 pat
LI R1,>FF      Start with blank pattern
BLWP @VSW    Store pattern byte in VRAM
LI R2,7      7 more pattern bytes
STRPAT MOV R1,@8000 Store pattern byte in VRAM
DEC R2      Next character pattern byte
JNE STRPAT
SRC R1,1    Next Character pattern
LI R2,8    8 pattern bytes per character
DEC R3    Next character
JNE STRPAT Last character?
LI R0,>390 VRAM location for ASCII 128 col
LI R1,>5000 Light blue/transparent colour
BLWP @VSW Colour to table in VRAM
MOV R1,@8000
*****
*Top of screen values*
*****
LI R2,26
LI R0,3
LI R1,TITLE
BLWP @VMBW Show title on top screen line
LI R1,MWS+32 Storage location for bar chars
INC R3
LI R0,35 Screen location for start of BAR
LI R9,>607 Joystick 1/joystick 2 select
LI R8,3000 Initial delay
*****
*Routine to show bar value on screen*
*****
SHWBAR MOV R3,R5
MOV R5,R6
SRL R6,3 Divide by 8
JEQ ENDCR At LHS of bar?
LI R7,>8800 Solid character pattern
BLOCK MOV R7,*R1+ Store bar character code
DEC R6

```

```

JNE BLOCK
ENDCR CI R1,BARS At RHS of bar?
JEQ FINBAR
MOV R6,R6 Flag for part character
JNE STRVAL
ANDI R5,7 Get remainder of divide by 8
AI R5,>80 Calculate character
SWPB R5
STRVAL MOV R5,*R1+ Store bar character code
LI R5,>8000 Blank character
SETO R6 Flag for blank character
JMP ENDCR
FINBAR S R2,R1
BLWP @VMBW Show bar on screen
MOV R8,R6 Get delay count
DELAY DEC R6
JNE DELAY
*****
*Setup for joystick sampling*
*****
JOYSTIK LI R12,>24 CRU address >0024(keybd select)
SWPB R9 Alternate joysticks
LDCR R9,3 Set for joystick sampling
LI R12,6 CRU address >0006(keybd lines)
STCR R5,5 Get joystick values
*The bit values in R5 at this point are UP , DOWN , LEFT
*RIGHT and FIRE in bits 3 , 4 , 5 , 6 and 7 respectively
SLA R5,4 Set to check UP/DOWN
JNC DDEC Up pressed?
JLT TLEFT Down pressed?
DINC INC R8 Increase delay
CI R8,30001 Delay high limit?
JNE TLEFT
DDEC DEC R8 Decrease delay
JEQ DINC Delay low limit?
TLEFT SLA R5,2 Set to check LEFT/RIGHT
JNC INC RIGHT pressed?
JGT DEC LEFT pressed?
SLA R5,1
JLT JOYSTIK FIRE pressed?
BLWP @O QUIT
DECR DEC R3 Lower bar value
JNE SHWBAR
INC INC R3 Increase bar value
CI R3,201 Value at high limit
JEQ DEC
JMP SHWBAR
END

```

END



## COMPUTER WAVE

325 George Street  
(Near Wynyard Station)  
Sydney  
Phone: (02) 29 1631

### PTY LIMITED

A comprehensive range of Texas Instruments  
Education, Games, and Personal Business  
cartridges + many 3rd party cassettes.

We also have TI-99/4(A) Peripherals and  
hardware like Modems, Monitors & Printers.

We specialise in cables for your hardware needs



Come up and see us some Time! **IMAGIC**



## PROGRAM \*1

```

100 !*****
110 ! CALENDARS *
120 ! EXTENDED BASIC RQD. *
130 ! MY FIRST PROGRAMME! *
    ! FIRST WRITTEN IN NOV*
    ! '83 REVISED MAR '84 *
    ! AND AGAIN JULY 1985.*
140 ! BY ALAN SWALES *
    ! FOR T.I.S.H.U.G. *
150 !*****
160 !
170 !
180 !
190 !
200 CALL CLEAR :: CALL SCREE
N(6):: FOR SET=0 TO 12 :: CA
LL COLOR(SET,1,1):: NEXT SET
210 DATA "JANUARY","FEBRUARY
","MARCH","APRIL","MAY","JUN
E","JULY","AUGUST","SEPTEMBE
R","OCTOBER","NOVEMBER","DEC
EMBER"
220 DATA 31,28,31,30,31,30,3
1,31,30,31,30,31
230 IMAGE ##
240 PRINT " C A L E N D
A R S": : : :
250 PRINT " BY A. SWALES
, 1984": : : :
260 PRINT " A PROGRAM TO
PRINT A CALENDAR OF AN
Y MONTH OF ANY YEAR B
ETWEEN 1753 AND 200
00 AD": : : :
270 PRINT "PRESS [ENTER]" ::
FOR SET=0 TO 12 :: CALL COL
OR(SET,16,1):: NEXT SET
280 CALL KEY(0,K,S)
290 IF (S=0)+(K<>13)THEN 280
ELSE CALL CLEAR
300 INPUT "DO YOU WISH TO PR
INT THE CALENDAR? (Y/N)
":ANS$ :: IF AN$<>
"Y" THEN 320
310 PRINT :: INPUT "WHICH MO
DE?...PIO OR RS232":PRINT$
:: OPEN #2:PRINT$,OUTPUT
320 CALL CLEAR :: PRINT " e
.g. MARCH 1985= (3,1985)
OR
(0,0) TO END PROGRAMME": :
: : :
330 INPUT "ENTER MONTH & YE
AR REQUIRED(M,Y)":M,Y
340 IF M=0 OR Y=0 THEN 880 !
TO END
350 DIM M$(12),D(12)!SETS UP
ARRAY
360 IF M<0 OR M>12 OR Y<1753
OR Y>20000 THEN 330 !TEST I
NPUT
370 IF M>2 AND Y-INT(Y/4)*4=
0 THEN L=1 ELSE L=0 !ADJUST
START OF COUNT IN LEAP YEAR
FOR 29 DAY FEB
380 IF Y-INT(Y/400)*400=0 TH
EN 400 !TEST FOR 4TH CENTURY
YEAR WHEN 29FEB DYS
390 IF Y-INT(Y/100)*100=0 TH
EN L=0 ELSE 400 !TEST FOR CE
NTURY YEARS FEB=28DAYS LY
400 CALL CLEAR :: MD=L !INIT
IALISE DAYCOUNT
410 RESTORE 220 :: MD=MD+1 !
CORRECTION FOR JAN 1 1753=MO
NDAY
420 FOR K=1 TO M-1 :: READ M
M :: MD=MD+MM :: NEXT K
430 RESTORE 210
440 FOR I=1 TO 12 :: READ M$
(I):: NEXT I !READ MONTH DAT
A
450 PRINT TAB(2);"MONTH OF "
;M$(M);Y;"AD" :: PRINT !NAME
OF MONTH

```

```

460 IF AN$="Y" THEN PRINT #2
:TAB(22);"MONTH OF ";M$(M);Y
;"AD" !PRINT IT ON PRINTER
470 RESTORE 220 !SET READ DA
TA TO ZERO
480 FOR H=1 TO M :: READ D(M
):: NEXT H !READ No. DAYS IN
MONTH DATA
490 N=D(M)
500 IF M=2 AND Y-INT(Y/4)*4=
0 THEN N=N+1 ELSE 530 !FEB 2
9 DAYS LY.(N)
510 IF M=2 AND Y-INT(Y/100)*
100=0 THEN N=N-1 ELSE 530 !C
ENT YR NO LEAP DY
520 IF Y-INT(Y/400)*400=0 TH
EN N=N+1 ELSE 530 !4TH CENT
INCLUDE LDAY
530 YDIF=Y-1752 :: LY=INT((Y
DIF-1)/4):: CY=INT(Y/100)-17
:: C4Y=INT(Y/400)-4 !LOOKS
FOR EXTRA DAYS LYS ETC
540 DT=365*YDIF+LY-CY+C4Y+MD
!TOTAL DAYS(DAY TOTAL)FROM
31DEC1752
550 S=DT-(INT(DT/7)*7)!SET P
RINT START DAY OF WEEK
560 K=1 :: J=1 !INIT K & J
570 IF Y-INT(Y/100)*100=0 TH
EN S=S+1 !IF CENT YR,28 DAY
FEB
580 IF Y-INT(Y/400)*400=0 TH
EN S=S-1 !IF 4TH CENT YR THE
N PUT 29 DYS FEB
590 IF S=2 THEN J=5 !ADJ PRN
T
600 IF S=3 THEN J=9
610 IF S=4 THEN J=13
620 IF S=5 THEN J=17
630 IF S=6 THEN J=21
640 IF S=7 THEN J=25
650 IF S=8 THEN J=1
660 IF S=0 THEN J=25
670 PRINT TAB(2);"S M T
W T F S" :: PRINT ::
PRINT
680 FOR I=K TO N :: PRINT TA
B(J):: PRINT USING 230:I::
J=J+4
690 IF J>26 THEN PRINT :: PR
INT :: PRINT :: J=1
700 NEXT I :: PRINT
710 IF Y-INT(Y/4)*4=0 AND Y-
INT(Y/100)*100<>0 OR Y-INT(Y
/400)*400=0 THEN PRINT Y;"=L
EAP YEAR"
720 IF AN$="Y" THEN 730 ELSE
870 :: J=21
730 IF S=1 THEN J=21
740 IF S=2 THEN J=25
750 IF S=3 THEN J=29
760 IF S=4 THEN J=33
770 IF S=5 THEN J=37
780 IF S=6 THEN J=41
790 IF S=7 THEN J=45
800 IF S=8 THEN J=21
810 IF S=0 THEN J=45 :: K=1
820 PRINT #2:TAB(22);"S M
T W T F S "
830 FOR I=K TO N :: PRINT #2
:TAB(J):: PRINT #2:I;
840 J=J+4 :: IF J>48 THEN PR
INT #2: : J=21
850 NEXT I
860 PRINT #2:
870 INPUT "DO YOU WANT ANOTH
ER MONTH? (Y/N)":Q$
880 IF Q$="Y" THEN 330 ELSE
END

```



## Nº2

```

100 REM *****
110 REM MAGIC MACHINE*****
120 REM *****
130 REM D.N.HARRIS*****
140 REM *****
150 REM PRESENTED 3/8/85*
160 REM *****
170 REM NEEDS TERMINAL***
180 REM EMULATOR II AND**
190 REM SPEECH
200 REM SYNTHESIZER*****
210 REM *****
220 IN=1
230 CALL CLEAR
240 OPEN #1:"SPEECH",OUTPUT
250 PRINT #1:"//50 -100"
260 FOR SET=1 TO 12
270 CALL COLOR(SET,14,11)
280 NEXT SET
290 RESTORE
300 READ A$
310 IN=ABS((30-LEN(A$))/2)
320 IF (IN/2-INT(IN/2))>0 TH
EN 330 ELSE 340
330 IN=IN-1
340 CALL SCREEN(IN)
350 A=LEN(A$)
360 ON A GOSUB 810,830,850,8
70,890,910,930,950,970,990,1
010,1030,1050,1070,1090,1110
,1130,1150,1170,1190,1210,12
30,1250,1270,1290,1310,1330,
1350
370 PRINT TAB(IN);A$
380 PRINT #1:A$
390 IF A$="SO MIGHT IT BE" T
HEN 290
400 GOTO 300
410 STOP
420 DATA A
430 DATA AB
440 DATA ABR
450 DATA ABRA
460 DATA ABRA
470 DATA ABRACA
480 DATA ABRACAD
490 DATA ABRACADA
500 DATA ABRACADAB
510 DATA ABRACADABR
520 DATA ABRACADABRA
530 DATA THE--MORE--THE--WIL
L
540 DATA THE-- LESS-- THE--
WORK
550 DATA AS-- ALL-- THE-- SP
IRITS
560 DATA LOVE-- TO-- SHIRK
570 DATA BUT-- AS-- THE-- PR
ACTICE
580 DATA GROW-- THE-- SKILL
590 DATA SO-- EVERY-- DAY
600 DATA SHALL-- MORE-- THE--
WILL
610 DATA SHOW-- LIFE-- TO-- LOV
ING
620 DATA DEEDS-- THAT-- SHOW
630 DATA THE--WAY-- ALL-- GOOD
640 DATA DOES--TEND--TO--GO
650 DATA TO--PUT--OFF--WORSE
660 DATA AND--PUT--ON--BETTE
R
670 DATA TO--SEND--A--USEFUL
680 DATA ONE--A--LETTER
690 DATA THE--MORE--THE--SKILL
700 DATA THE--LESS--THE--WORK
710 DATA SO--THOUGH--THE--SPIRI
TS
720 DATA LOVE--TO--SHIRK
730 DATA THUS--SHALL--IT--BE
740 DATA AS--DONE--BY--ME
750 DATA ORDER--MORE--AND--CHA
OS--LESS
760 DATA UNTIL--MY--HABIT'
S

```

```

770 DATA NO>^DISTRESS
780 DATA AND ALL MY DEEDS
790 DATA ARE SWEET SUCCESS
800 DATA SO MIGHT IT BE
810 PRINT #1:"//1 3.2"
820 RETURN
830 PRINT #1:"//2 6.4"
840 RETURN
850 PRINT #1:"//3 9.6"
860 RETURN
870 PRINT #1:"//4 12.8"
880 RETURN
890 PRINT #1:"//5 16"
900 RETURN
910 PRINT #1:"//6 19.2"
920 RETURN
930 PRINT #1:"//7 22.4"
940 RETURN
950 PRINT #1:"//8 25.6"
960 RETURN
970 PRINT #1:"//9 28.8"
980 RETURN
990 PRINT #1:"//10 32"
1000 RETURN
1010 PRINT #1:"//11 35.2"
1020 RETURN
1030 PRINT #1:"//12 38.4"
1040 RETURN
1050 PRINT #1:"//13 41.6"
1060 RETURN
1070 PRINT #1:"//14 44.8"
1080 RETURN
1090 PRINT #1:"//15 48"
1100 RETURN
1110 PRINT #1:"//16 51.2"
1120 RETURN
1130 PRINT #1:"//17 54.4"
1140 RETURN
1150 PRINT #1:"//18 57.6"
1160 RETURN
1170 PRINT #1:"//19 60.8"
1180 RETURN
1190 PRINT #1:"//20 64"
1200 RETURN
1210 PRINT #1:"//21 67.2"
1220 RETURN
1230 PRINT #1:"//22 70.4"
1240 RETURN
1250 PRINT #1:"//23 73.6"
1260 RETURN
1270 PRINT #1:"//24 76.8"
1280 RETURN
1290 PRINT #1:"//25 80"
1300 RETURN
1310 PRINT #1:"//26 83.2"
1320 RETURN
1330 PRINT #1:"//27 86.4"
1340 RETURN
1350 PRINT #1:"//28 89.6"
1360 RETURN
100 REM*****
110 REM* MICROWAVE TIME/ *
120 REM*WEIGHT CALCULATOR*
130 REM* BY LES TUTCHINGS*
140 REM* FROM TIBUG *
150 REM* X/B REQUIRED *
160 REM*****
170 CALL CLEAR :: ON WARNING
NEXT :: CALL SCREEN(12):: X
$="MICROWEIGHT" :: X=LEN(X$)
:: DISPLAY AT(12,(28-X)/2)BE
EP:X$
180 FOR DE=1 TO 400 :: NEXT
DE :: DISPLAY AT(1,3)ERASE A
LL:"ENTER DATA AS REQUESTED"
: : : " WEIGHT FACTOR (GRAMS)
500"
190 DISPLAY AT(9,2):"TIME FA
CTOR (MINUTES)": : : : " WE
IGHT OF ITEM(GRAMS)"
200 ACCEPT AT(4,24)VALIDATE(
NUMERIC)SIZE(-5)BEEP:WF :: A
CCEPT AT(9,24)VALIDATE(NUMER
IC):TF :: ACCEPT AT(14,24)VA
LIDATE(NUMERIC)BEEP:WI

```

```

210 NT=(TF/WF)*WI :: DISPLAY
AT(19,2):"COOKING TIME IS";
NT;"MINUTES" :: DISPLAY AT(2
4,1)BEEP:"ANY MORE DATA WANT
ED (Y/N)?"
220 CALL KEY(3,K,S):: IF S=0
THEN 220 ELSE IF K=89 THEN
200 ELSE CALL CLEAR :: END

```



```

100 REM *****
110 REM * BOGGLED *
120 REM * BY BATZ *
130 REM * IN THE ATICC*
140 REM * 13.12.84 *
150 REM * BOTH BASICS *
160 REM *****
170 CALL CLEAR
180 CALL SCREEN(16)
190 PRINT TAB(12);"BOGGLED":
: : :
200 PRINT TAB(14);"BY": : :
: :
210 PRINT TAB(7);"BATZ IN TH
E ATTICC": : : :
220 PRINT "DO YOU REQUIRE IN
STRUCTIONS?": : :
230 PRINT TAB(12);"(Y/N)"
240 CALL KEY(3,K,S)
250 IF S=0 THEN 240
260 IF K=89 THEN 1320
270 IF K<>78 THEN 240
280 CALL CLEAR
290 TM=180
300 XX=1
310 V=9
320 H=13
330 D=26
340 CALL COLOR(11,2,1)
350 FOR ST=1 TO 8
360 CALL COLOR(ST,6,6)
370 NEXT ST
380 CALL CHAR(42,"007E7E7E7E
7E7E")
390 CALL CHAR(81,"6090959595
B29864")
400 CALL CHAR(97,"0000001F1F
18181818")
410 CALL CHAR(98,"000000F8F8
181818")
420 CALL CHAR(99,"1818181F1F
")
430 CALL CHAR(100,"181818F8F
8")
440 CALL CHAR(101,"000000FFF
F")
450 CALL CHAR(102,"181818181
8181818")
460 CALL CHAR(103,"181818FFF
F181818")
470 CALL CHAR(104,"181818FFF
F")
480 CALL CHAR(105,"000000FFF
F181818")
490 CALL CHAR(106,"181818F8F
8181818")
500 CALL CHAR(107,"1818181F1
F181818")
510 CALL CHAR(108,"0")
520 CALL CHAR(112,"007E7E7E7
E7E7E")
530 PRINT : : : : :
540 PRINT TAB(10);"aeicieieb
"
550 PRINT TAB(10);"f1f1f1f1f
"
560 PRINT TAB(10);"kegegegej
"
570 PRINT TAB(10);"f1f1f1f1f
"
580 PRINT TAB(10);"kegegegej
"
590 PRINT TAB(10);"f1f1f1f1f
"
600 PRINT TAB(10);"kegegegej
"

```

```

610 PRINT TAB(10);"f1f1f1f1f
"
620 PRINT TAB(10);"cehehehed
"
630 PRINT : : :
640 PRINT TAB(5);"0";TAB(11)
;"1";TAB(17);"2";TAB(23);"3"
650 PRINT : : :
660 CALL HCHAR(22,8,42,18)
670 PRINT TAB(10);"TIME LEFT
"
680 DIM US(30),LR(30)
690 FOR Z=1 TO 16
700 FOR I=1 TO 6
710 READ A(I)
720 NEXT I
730 AB=INT(RND*6)+1
740 LR(Z)=A(AB)
750 NEXT Z
760 RANDOMIZE
770 AA=INT(RND*16)+1
780 US(XX)=AA
790 FOR CY=0 TO XX-1
800 IF US(CY)=US(XX)THEN 770
810 NEXT CY
820 CH=LR(AA)
830 CALL HCHAR(V,H,CH)
840 H=H+2
850 IF H<=19 THEN 890
860 H=13
870 IF H<>13 THEN 890
880 V=V+2
890 XX=XX+1
900 IF XX<17 THEN 770
910 CALL HCHAR(21,7,112)
920 FOR ST=1 TO 8
930 CALL COLOR(ST,16,1)
940 NEXT ST
950 CALL SCREEN(14)
960 CALL HCHAR(22,8,101,18)
970 TM=TM-1
980 AA=TM/10
990 IF AA=INT(AA)THEN 1120
1000 FOR DEL=1 TO 270
1010 NEXT DEL
1020 IF TM>0 THEN 970
1030 CALL COLOR(11,10,1)
1040 CALL SOUND(999,600,0)
1050 PRINT " YOUR TIME
IS UP": :
1060 PRINT "PRESS E TO EXIT"
:"PRESS ANY KEY TO PLAY AGAI
N"
1070 CALL KEY(3,K,S)
1080 IF S=0 THEN 1070
1090 IF K=89 THEN 1570
1100 RESTORE
1110 GOTO 280
1120 D=D-1
1130 CALL HCHAR(21,D,32)
1140 GOTO 1000
1150 DATA 73,65,84,89,66,76
1160 DATA 65,77,79,81,66,74
1170 DATA 76,69,75,85,89,71
1180 DATA 87,71,76,82,85,73
1190 DATA 85,68,79,78,84,85
1200 DATA 82,79,70,73,66,88
1210 DATA 77,79,82,65,72,83
1220 DATA 82,65,76,69,67,83
1230 DATA 69,83,85,84,76,80
1240 DATA 69,72,89,70,69,73
1250 DATA 77,80,68,67,69,65
1260 DATA 65,73,79,65,84,67
1270 DATA 80,73,78,69,83,72
1280 DATA 86,69,90,65,68,78
1290 DATA 84,73,86,69,71,78
1300 DATA 83,78,87,68,79,69
1310 GOTO 280
1320 CALL CLEAR
1330 PRINT TAB(5);"BOGGLED I
NSTRUCTIONS"
1340 PRINT TAB(5);"*****
*****": :

```



```

1350 PRINT "WHEN THE GAME STARTS, EACH PLAYER SEARCHES THE LETTERS ON THE BOARD FOR WORDS OF 3"
1360 PRINT "OR MORE LETTERS. WHEN A WORD IS FOUND BY A PLAYER, HE/SHE WRITES IT DOWN."
1370 PRINT "THE WORDS ARE FORMED FROM ADJOINING LETTERS. LETTERS MUST JOIN IN THE PROPER"
1380 PRINT "SEQUENCE, HORIZONTALLY VERT-CALLY OR DIAGONALLY, TO THE LEFT RIGHT, OR UP AND DOWN."
1390 PRINT "NO LETTER MAY BE USED MORE THAN ONCE IN EACH WORD."
1400 PRINT : : "(PRESS ANY KEY TO CONTINUE)"
1410 GOSUB 1590
1420 PRINT "ANY WORD, PROVIDED THAT IT CAN BE FOUND IN A STANDARD ENGLISH DICTIONARY MAY BE"
1430 PRINT "USED. WORDS WITH IN WORDS ARE ALSO PERMISSIBLE; E.G. -SPARE spa, par, sre, spar, pare."
1440 PRINT "WHEN THE TIMER RUNS OUT, EVERYONE STOPS WRITING. EACH PERSON THEN READS HIS/HER"
1450 PRINT "LIST ALOUD, AND ANY WORD APPEARING ON MORE THAN ONE PLAYER'S LIST IS CROSSED OFF"
1460 PRINT "ALL THE LISTS.": "AFTER ALL THE PLAYERS HAVE READ THEIR LISTS, EACH PERSON THEN SCORES THE"
1470 PRINT "REMAINING WORDS: -: "NO OF LETTERS: 3 4 5 6 7 8+
-----"
1480 PRINT "POINTS: 1
1 2 3 5 11": "(PRESS ANY KEY TO CONTINUE)"
1490 GOSUB 1590
1500 PRINT "THE WINNER IS THE PLAYER WHOSE SCORE IS THE HIGHEST OR WHO FIRST REACHES AN"
1510 PRINT "AGREED TARGET SCORE, 50, 100 POINTS ETC.": "MULTIPLE MEANINGS OF THE SAME WORD DO NOT EARN EXTRA"
1520 PRINT "CREDIT.": "THE SAME WORD FOUND IN A DIFFERENT AREA DOES NOT COUNT FOR EXTRA CREDIT":
1530 PRINT "QU COUNTS AS TWO LETTERS": "SINGULAR AND PLURAL FORMS EARN CREDIT PROVIDED THAT"
1540 PRINT "THEY ARE LISTED AS SEPARATE WORDS.": TAB(3); "(PRESS ANY KEY TO START)"
1550 GOSUB 1590
1560 GOTO 280
1570 CALL CLEAR
1580 STOP
1590 CALL KEY(3,K,S)
1600 IF S=0 THEN 1590
1610 CALL CLEAR
1620 RETURN

```



```

100 ! *****
110 ! *AIR TRAFFIC CONTROL*
120 ! *****
130 ! TISHUG LIBRARY 18&19
140 ! EXTENDED BASIC
150 DISPLAY AT(1,1)ERASE ALL : "YOU ARE IN COMMAND OF A BUSY"
160 DISPLAY AT(2,1): "AIRPORT TERMINAL. FLIGHTS"
170 DISPLAY AT(3,1): "WILL ARRIVE IN A HOLDING"
180 DISPLAY AT(4,1): "PATTERN, YOU MUST LAND THEM"
190 DISPLAY AT(5,1): "ON EITHER OF THREE RUNWAYS,"
200 DISPLAY AT(6,1): "WHICH MAY OR MAY NOT BE"
210 DISPLAY AT(7,1): "OPEN. EACH RUNWAY HAS A"
220 DISPLAY AT(8,1): "DIFFERENT FLIGHT PATH FOR"
230 DISPLAY AT(9,1): "APPROACH AND TAKEOFF."
240 DISPLAY AT(10,1): "ALSO YOU MAY CHANGE"
250 DISPLAY AT(11,1): "EACH PLANES ALTITUDE"
260 DISPLAY AT(12,1): "AND CHECK WHICH PLANES ARE"
270 DISPLAY AT(13,1): "NOT IN A HOLDING PATTERN."
280 DISPLAY AT(14,1): "PLANES WILL TAKEOFF"
290 DISPLAY AT(15,1): "AUTOMATICALLY AFTER FLIGHT"
300 DISPLAY AT(16,1): "NUMBER 8. "
310 DISPLAY AT(17,1): "THE COMPUTER WILL WARN OF"
320 DISPLAY AT(18,1): "CONFLICTS ON LANDINGS,"
330 DISPLAY AT(19,1): "BUT TAKEOFFS CANNOT BE"
340 DISPLAY AT(20,1): "ABORTED OR STOPPED."
350 DISPLAY AT(24,1)BEEP: "PRESS ANY KEY TO CONTINUE"
360 CALL KEY(0,KEYIN,STATE): IF STATE=0 THEN 360
370 DISPLAY AT(1,5)ERASE ALL : "* AIR TRAFFIC CONTROLLER *"
380 DISPLAY AT(3,5): "* BY JOHN J. VOLK *"
390 DISPLAY AT(7,5): "COMMANDS ARE : "
400 DISPLAY AT(9,5): "L-LAND PLANE"
410 DISPLAY AT(11,5): "A-CHANGE ALTITUDE"
420 DISPLAY AT(13,5): "C-ABORT LANDING"
430 DISPLAY AT(14,5): "* NOTE * ONLY ON LANDING"
440 DISPLAY AT(16,5): "T-TAKE OFF FROM TERMINAL"
450 DISPLAY AT(18,5): "S-CHECK STATUS"
460 DISPLAY AT(24,1)BEEP: "PRESS ANY KEY"
470 CALL KEY(0,KEYIN,STATE): IF STATE=0 THEN 470 ELSE 480
480 DIM T(28)
490 DISPLAY AT(2,1)ERASE ALL : "INPUT LEVEL OF DIFFICULTY"
500 DISPLAY AT(4,1): "1-TRAINEE"
510 DISPLAY AT(6,1): "2-CONTROLLER"
520 DISPLAY AT(8,1): "3-SENIOR CONTROLLER"
530 DISPLAY AT(10,1): "YOUR CHOICE ?"

```

```

540 ACCEPT AT(10,15)BEEP VALIDATE(NUMERIC,"123")SIZE(1): CHOICE
550 IF CHOICE=1 THEN SKILL=20
560 IF CHOICE=2 THEN SKILL=15
570 IF CHOICE=3 THEN SKILL=10
580 CALL CLEAR
590 CALL SCREEN(16)
600 DISPLAY AT(1,1): "5000"
610 DISPLAY AT(5,1): "4000"
620 DISPLAY AT(10,1): "3000"
630 DISPLAY AT(15,1): "2000"
640 DISPLAY AT(20,1): "1000"
650 CALL CHAR(96,"000018FF24")
660 CALL CHAR(97,"603098FFFF983060")
670 CALL CHAR(98,"060C19FFFF190C06")
680 CH=97 : SP=0 : VOL=30 : SCORE=0 : E=0 : C=1 : X=0 : T(X)=0
690 RANDOMIZE
700 VOL=30
710 IF SP>28 THEN 1900
720 IF X>=28 THEN X=0 : T(X)=0 : IF Z>=28 THEN Z=0
730 J=1+INT(SKILL*RND)
740 IF J=7 THEN 750 ELSE GOTO 940
750 REM FLIGHT VALUES
760 YPOS=1+INT(150*RND)
770 XPOS=1+INT(250*RND)
780 COLOR=2+INT(14*RND)
790 IF COLOR=16 OR COLOR=15 THEN 780
800 SP=SP+1
810 CALL SPRITE(#SP,96,COLOR,YPOS,XPOS)
820 J=1+INT(2*RND)
830 IF J=1 THEN 840 ELSE 850
840 CALL PATTERN(#SP,98): C ALL MOTION(#SP,0,-2): GOTO 860
850 CALL PATTERN(#SP,97): C ALL MOTION(#SP,0,2)
860 CALL SOUND(-4250,-3,30,122,20,118,20)
870 DISPLAY AT(23,1): "FLIGHT"; SP: "ARRIVING"
880 IF CH=98 THEN CALL MOTION(#SP,0,-2)
890 IF SP>8 THEN 900 ELSE 940
900 CALL COINC(ALL,V): IF V=-1 THEN CALL SOUND(-1000,-5,0): GOTO 1590
910 RANDOMIZE
920 OFF=1+INT(10*RND)
930 IF OFF=5 THEN GOSUB 1730
940 CALL COINC(ALL,V): IF V=-1 THEN CALL SOUND(-1000,-5,0): GOTO 1590
950 CALL KEY(0,KEY,ST)
960 DISPLAY AT(24,1): "** COMMAND ? **"
970 IF ST=0 THEN 1020
980 IF KEY=65 THEN GOSUB 1050 ! CHANGE ALTITUDE
990 IF KEY=76 THEN GOSUB 1220 ! LAND PLANE
1000 IF KEY=84 THEN GOSUB 1730 ! TAKEOFF
1010 IF KEY=83 THEN GOSUB 1680 ! CHECK STATUS
1020 CALL COINC(ALL,V)
1030 IF V=-1 THEN CALL SOUND(1000,-5,0): GOTO 1590
1040 GOTO 690
1050 DISPLAY AT(23,1): "WHICH FLIGHT ?"
1060 ACCEPT AT(23,24)BEEP VALIDATE(NUMERIC)SIZE(2): ANE

```

```

1070 IF ANE>SP THEN DISPLAY
AT(23,1)BEEP:"INVALID FLIGHT
" :: GOTO 690
1080 FOR K=Z TO SP
1090 IF ANE=T(K)THEN DISPLAY
AT(23,1):"INVALID FLIGHT" :
: GOTO 690
1100 NEXT K
1110 DISPLAY AT(24,1):"ALTIT
UDE + OR - IN FEET"
1120 ACCEPT AT(24,24)BEEP VA
LIDATE(NUMERIC)SIZE(5):ALTIT
UDE
1130 ON WARNING NEXT
1140 ON ERROR 690
1150 ALTITUDE=INT((ALTITUDE/
10)/3.0)
1160 CALL SOUND(-900,-6,10,1
22,20,118,20)
1170 CALL POSITION(#ANE,YPIT
,XPIT)
1180 CALL LOCATE(#ANE,(YPIT-
ALTITUDE),XPIT)
1190 CALL COINC(ALL,V)
1200 IF V=-1 THEN CALL SOUND
(3000,-5,0):: GOTO 1590
1210 RETURN
1220 DISPLAY AT(23,1):"WHICH
FLIGHT ?"
1230 ACCEPT AT(23,24)BEEP VA
LIDATE(NUMERIC)SIZE(2):M
1240 ON WARNING NEXT
1250 ON ERROR 950
1260 IF M>SP THEN DISPLAY AT
(23,1):"*** INVALID FLIGHT *"
:: GOTO 940
1270 FOR K=0 TO SP
1280 IF M=T(K)THEN DISPLAY A
T(23,1):"*** INVALID FLIGHT *"
:: GOTO 940
1290 NEXT K
1300 DISPLAY AT(24,1):"WHICH
RUNWAY (1-2-3) ?"
1310 ACCEPT AT(24,24)BEEP VA
LIDATE(NUMERIC,"123")SIZE(1)
:R
1320 IF R>=4 THEN 1300
1330 RANDOMIZE
1340 J=1+INT(3*RND)
1350 IF J=1 THEN DISPLAY AT(
23,1):"RUNWAY BLOCKED-CHOOSE
AGAIN" :: GOTO 1310
1360 CALL PATTERN(#M,96)
1370 IF R=1 THEN CALL MOTION
(#M,2,-1)
1380 IF R=2 THEN CALL MOTION
(#M,2,0)
1390 IF R=3 THEN CALL MOTION
(#M,2,1)
1400 DISPLAY AT(23,1):"FLIGH
T ";M;"NOW LANDING"
1410 CALL POSITION(#M,YDIR,X
DIR)
1420 CALL COINC(ALL,V):: IF
V=-1 THEN CALL SOUND(-1000,-
5,0):: GOTO 1590
1430 IF VOL<=0 THEN VOL=0 EL
SE VOL=VOL-.5 :: CALL SOUND(
-4250,-6,VOL,122,20,118,20)
1440 IF YDIR>180 THEN 1580
1450 IF YDIR>=140 THEN GOTO
1530
1460 FOR B=1 TO SP
1470 IF B=T(X)THEN GOTO 1510
1480 IF B=M THEN GOTO 1510
1490 CALL DISTANCE(#B,#M,DI)
1500 IF SQR(DI)>=40 THEN GOT
O 1520 ELSE DISPLAY AT(24,1)
BEEP:"*** WARNING CONFLICT *"
"
1510 FOR DELAY=1 TO 10 :: NE
XT DELAY
1520 NEXT B
1530 DISPLAY AT(24,1):""

```

```

1540 CALL COINC(ALL,V):: IF
V=-1 THEN CALL SOUND(-1000,-
5,0):: GOTO 1590
1550 CALL KEY(0,K1,S):: IF K
1=67 THEN CALL PATTERN(#M,97
):: CALL MOTION(#M,0,2):: RE
TURN
1560 IF YDIR>=180 THEN 1570
ELSE 1410
1570 CALL LOCATE(#M,255,C)::
C=C+8
1580 CALL DELSPRITE(#M):: T(
X)=M :: SCORE=SCORE+20 :: X=
X+1 :: DISPLAY AT(23,1):""
" ::
RETURN
1590 REM COLLISION
1600 RANDOMIZE
1610 CALL DELSPRITE(ALL)
1620 PEOPLE=99+INT(900*RND)
1630 DISPLAY AT(12,5)BEEP ER
ASE ALL:"MID-AIR COLLISION.
";PEOPLE;"HUMAN LIVES LOST"
1640 DISPLAY AT(16,5):"YOUR
SCORE IS ";SCORE
1650 DISPLAY AT(24,1):"PLAY
AGAIN (Y/N) Y"
1660 ACCEPT AT(24,18)BEEP VA
LIDATE(UALPHA,"YN")SIZE(-1):
ANSWER$
1670 IF ANSWER$="Y" THEN 480
ELSE END
1680 FOR STAT=0 TO X
1690 DISPLAY AT(24,1)BEEP:"F
LIGHTS NOT IN HOLD ";T(STAT)
1700 FOR DELAY=1 TO 50 :: NE
XT DELAY
1710 NEXT STAT
1720 RETURN
1730 REM TAKEOFF
1740 E=T(Z)
1750 IF E=0 THEN RETURN
1760 Z=Z+1 :: IF Z>28 THEN Z
=0
1770 DISPLAY AT(23,1):"FLIGH
T ";E;"READY FOR TAKEOFF"
1780 VOL=0
1790 DISPLAY AT(24,1):"RUNWA
Y (1-2-3) ?"
1800 ACCEPT AT(24,18)BEEP VA
LIDATE(NUMERIC,"123")SIZE(1)
:WAY
1810 RANDOMIZE :: Y=1+INT(3*
RND):: IF Y=1 THEN DISPLAY A
T(24,1):"RUNWAY UNAVAILABLE"
:: GOTO 1770
1820 CALL SPRITE(#E,96,COLOR
,190,128)
1830 IF WAY=1 THEN CALL MOTI
ON(#E,-2,-1)
1840 IF WAY=2 THEN CALL MOTI
ON(#E,-2,0)
1850 IF WAY=3 THEN CALL MOTI
ON(#E,-2,1)
1860 CALL POSITION(#E,F,G)
1870 CALL SOUND(-900,-7,VOL,
122,VOL,118,VOL):: VOL=VOL+.
5 :: IF VOL>=20 THEN VOL=20
1880 IF F<=10 THEN CALL DELS
PRITE(#E):: SCORE=SCORE+40 :
: RETURN
1890 CALL COINC(ALL,V):: IF
V=-1 THEN CALL SOUND(3000,-5
,0):: GOTO 1590 ELSE 1860
1900 DISPLAY AT(12,1)BEEP:"Y
OUR SCORE IS ";SCORE
1910 DISPLAY AT(16,1):"PLAY
AGAIN ? (Y/N)"
1920 ACCEPT AT(16,20)BEEP VA
LIDATE(UALPHA,"YN"):ANSWER$
1930 IF ANSWER$="Y" THEN GOT
O 480 ELSE END

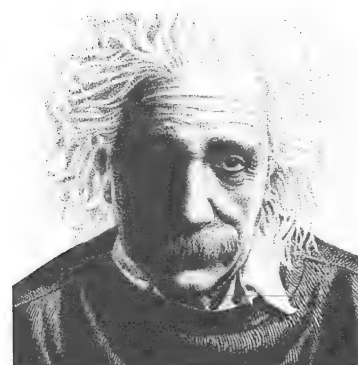
```

```

100 REM *****
110 REM * 28 COLUMN LISTER*
120 REM *FROM FILES LISTED*
130 REM *TO"DSK1.FILENAME"*
140 REM * JIM PETERSON *
150 REM *TIGERCUB SOFTWARE*
160 REM *****
170 DISPLAY AT(2,4)ERASE ALL:"28-COLUMN
CONVERTER" :: DISPLAY AT(5,12):"by Jim P
eterson"
180 DISPLAY AT(7,1):" To convert a progr
am, saved","with LIST "DSK1.FILENAME""
,"into 28-column format which","can be
merged into the text"
190 DISPLAY AT(11,1):"buffer of TI-Write
r."
200 DISPLAY AT(13,1):" Optionally with t
ransliter-","ated @, & and % for correct
","printing from formatter":"mode."
210 DISPLAY AT(18,1):" Do you want to pr
int the":"file from the":"(E)ditor?":"
(F)ormatter?"
220 ACCEPT AT(23,1)VALIDATE("EF")BEEP:Q$
230 DIM A$(1000):: CALL CLEAR :: INPUT "
What is the FILENAME? DSK1." :FN$ :
: FN$="DSK1."&FN$ :: PRINT ::
240 INPUT "What is the new FILENAME ? "
:PN$ :: PN$="DSK1."&PN$ :: OPEN #1:FN$,D
ISPLAY ,VARIABLE 80,INPUT :: OPEN #2:PN$
,DISPLAY ,VARIABLE 80,OUTPUT
250 IF Q$="E" THEN 290
260 PRINT #2:".TL 92:94;"
270 PRINT #2:".TL 91:64;"
280 PRINT #2:".TL 93:38;"
290 FOR L=1 TO 1000 :: LINPUT #1:A$(L)
300 IF LEN(A$(L-1))=80 OR LEN(A$(L-1))=1
60 THEN A$(L-1)=A$(L-1)&A$(L): L=L-1
310 IF EOF(1)THEN L=L+1 :: GOTO 330
320 NEXT L
330 FOR J=1 TO L-1
340 S=1
350 FOR T=1 TO 10
360 B$(T)=SEG$(A$(J),S,28)
370 IF Q$="E" THEN 390
380 GOSUB 460
390 S=S+28
400 NEXT T
410 FOR N=1 TO 10
420 IF B$(N)<>"" THEN PRINT #2:B$(N)
430 NEXT N
440 NEXT J
450 CLOSE #2 :: CLOSE #1 :: END
460 DATA @,!,$,!,%,^,^
470 RESTORE 460
480 FOR W=1 TO 3
490 READ CH$,R$
500 X=POS(B$(T),CH$,1):: IF X=0 THEN 520
510 B$(T)=SEG$(B$(T),1,X-1)&R$&SEG$(B$(T
),X+1,LEN(B$(T)))
520 NEXT W
530 RETURN

```

\*



\*

Hi gang! Let's get going with yet another edition of the SND Younger Set.

I must say that I am most impressed with the response to the Front Cover Competition. It is obvious that we have some definite artistic talent within our clubs' Younger Set members. Unfortunately though, with all competitions there can only be one winner and that winner for our competition is:

Paul from Perthville.

Well done Paul, you will be receiving your module of Miner 2049er in the mail very soon.

I'm sure that all of you agree that Paul has put a lot of effort into his entry for our competition. Other entries will be featured on the front cover of our magazine in future issues so don't be dis-heartened.

The race for second prize was so close that after speaking to Shane I have decided to award two second prizes. These prizes will be going to:

Sam Mudie, and  
Matthew Harribah.

Bath Sam and Matthew will be receiving a cassette full of specially selected software.

I also received in the mail two very good cartoons from Todd Winterford which I mentioned in last months issue.

Thanks for the good work Todd.

Also in my mail bag was a letter from Peter Lees of Peakhurst which included a drawing of 'GRUG', a Pelloandethal Man pictured below.



This is Grug a  
Pelloandethal Man.

Peter suggested that we have a separate Hall of Fame section devoted to Tunnels of Doom. Unfortunately with all the work that I have to do it would be a little too much, but what I will do is include in the present Hall of Fame a few changes suggested by Peter.

Tunnels of Doom will now include all of the scoring information that is included on the screen. This includes such things as gold, experience points and level.

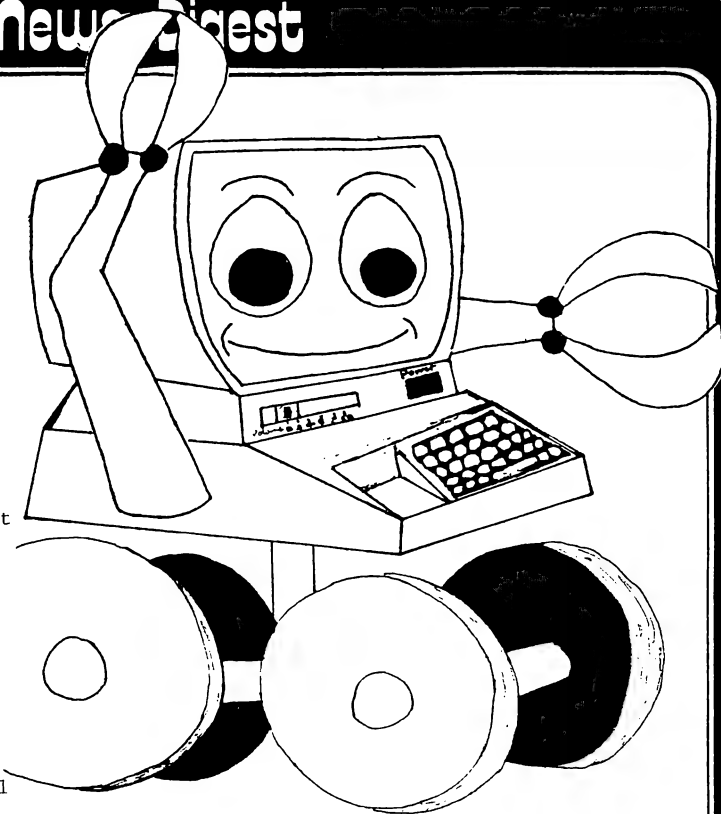
Peter also asked if there were any other games for the Tunnels of Doom cartridge besides Pennies & Prizes and Quest of the King and if there is a TI version of Lord British's Commodore 64 series ULTIMA games.

Well Peter, as far as I know these are the only games that have been written for the Tunnels of Doom. In regards to the ULTIMA games I haven't heard of any TI versions but I will speak to Keir Wells, our Review Columnist, and see if he can find out for you.

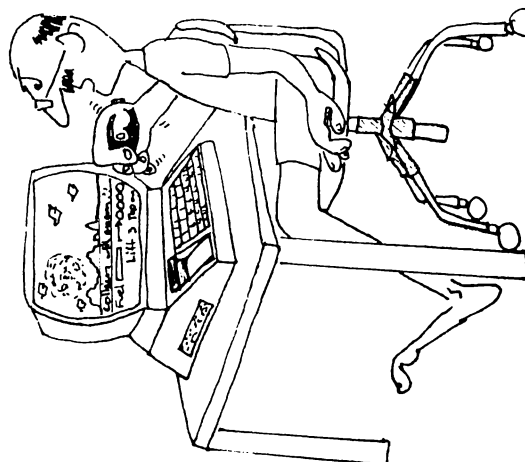
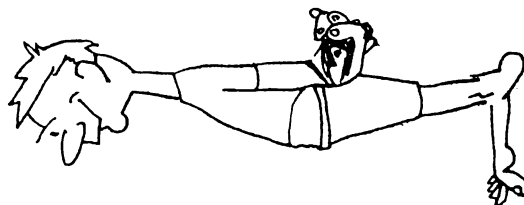
A PS on Peters' letter mentioned that his top score on BLASTO is 34,710. Keep blasting Peter.

Bye for now and keep sending in your letters.

*Jenny*  
JENNY.

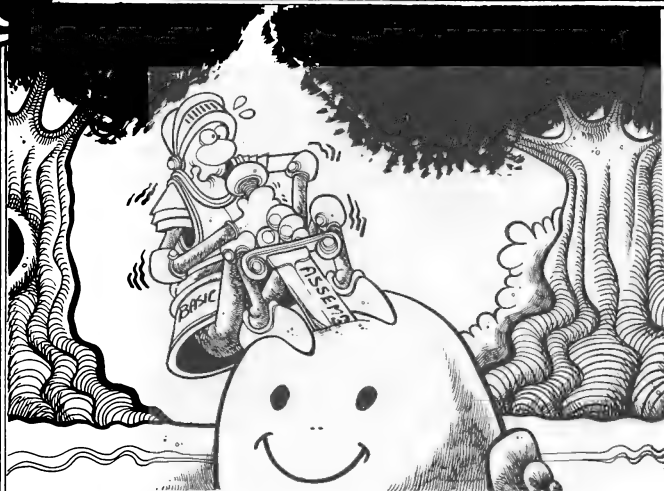


TEXAS INSTRUMENTS BUILDS  
PORTABLE COMPUTERS



DON'T WORRY DAD YOU'LL PROBABLY SHOT ONE EVENTUALLY





## CONVERSION ROUTINES.

by Peter HANCOCK (TICHUG)

The following subroutines may be of use to some members (especially those who use assembly language). They allow simple conversion of memory addresses and contents between hexadecimal and decimal notation.

Subroutine DEC is called as follows...

```
1000 READ HA$,HV$
1010 CALL DEC(HA$,DA)
1020 CALL DEC(HV$,DV)
1030 CALL LOAD(DA,DV)
1040 GOTO 1000
1050 DATA 3CD4,FF,3C00,01,30
00,2F...
```

Here HA\$ and HV\$ are string variables containing the memory address and value to be converted (e.g. "3CD4"). The subroutine returns the converted value in variables DA and DV respectively. If a value is greater than 32767, it is converted to a negative number as required by the CALL LOAD subroutine.

To disable this feature for both HEX and DEC subroutines and use them for straight conversions, remove the underlined statements.

Subroutine HEX is called as follows...

CALL HEX(D,HEX\$)

## LETTER TO THE EDITOR

"Respected TI.S.H.U.G. MEMBERS

I observe that your lack of initiative is being panned by your leaders and am offering you a chance to get into better spirits literally by means of a speech output program that exorcises lazy spirits and attracts ingenious kindness instead."

"..... I have found the old triangle of letters effective possibly because it suggests order and system and the alledged power seems like the Latin word 'Abrasadere' - to abrade -so the principle of 'elbow grease and approaching any job as a series of small steps to be added to is strongly implied.....The beginning of any work is to establish a border between the sort of thing the job makes more of and the sort of thing the job makes less of.

..... This computer program really does exorcise as you will discover by the disappearance of evil out of events surrounding its use..... The thought of

D is a variable containing the decimal value to be converted and HEX\$ is the hexadecimal string which results from the conversion. Again if D is negative, the subroutine will add 65536 to it for compatibility with LOAD and PEEK.

It would have been nicer if TI allowed multi-line user-defined functions. You could then produce functions to do nice things like...

CALL LOAD(DEC("3CD4"),DEC("FF"))

which fairly clearly shows that you want to place the hex value FF into location 3CD4. The current method of converting hex values into decimal and using these values as arguments makes programs totally incomprehensible.

As written they will only work for Extended Basic, but could easily be turned into GOSUBable routines for console Basic.

NOTE: To improve the speed of the routines, no checking is done on the validity of the input arguments so invalid inputs will simply produce the wrong results

```
30000 REM Convert hexadecimal
address in HEX$ into decimal
address in D.
30010 SUB DEC(HEX$,D)
30020 D=0
30030 FOR I=1 TO LEN(HEX$)
30040 D=16*D+POS("01234567
89ABC DEF",SEG$(HEX$,I,1), 1
)-1
30050 NEXT I
30060 IF D>32767 THEN D=D-65536
30070 SUBEND
```

```
31000 REM Convert decimal ad
dress in D into hexadecimal
address in HEX$.
31010 SUB HEX(D,HEX$)
31020 HEX$="":D1=D::IF D< 0
THEN D1=D1+65536
31030 HEX$=SEG$("0123456789A
BCDEF",D1-16*INT(D1/16)+1,1)
&HEX$
31040 D1=INT(D1/16):: IF D1<
>0 THEN 31030
31050 SUBEND
```

MANY THANKS PETER!

hard work as a means of success and endless striving for perfection as a delight in itself is apt to repel such phantoms as are mentioned in 'The Book of Common Prayer'.

The use of Speech Output for poetry generally is something that members of TI.S.H.U.G. seem to have overlooked and for those lacking in inspiration lots of published poems are just begging for the chance of the precise utterance a computer enables.....

After this I do not expect to see any complaints about lack of INSPIRATION among TI.S.H.U.G. members!"

Respectfully,

D.N. HARRIS - HURSTVILLE -24th July 1985.

COMMENT: Thanks for your support. I hope you don't mind this slightly edited version of your letter. I have tried to retain your BASIC message. I certainly hope your suggestion 'fuels the flames' of all that latent talent out there! ED.

## SECRETARY'S NOTES with J.R.

Hi! I have just returned from a five week visit to the UK. I was able to contact a number of TI enthusiasts, who are members of the TI99/4A Exchange Club and the TI Home Computer Users Club. If any members would like to join either of these organisations they can be reached at the following addresses:

TI99/4a EXCHANGE  
40 Barrhill, Patcham, Brighton, Sussex, BN1 8UF.  
Attention: Clive Scally.

TI Home Computer Users Club  
PO Box 190, Maidenhead, Berks SL6 1YX.  
Attention: Katie Lomax.

The second club has around 4000 members and originated from Paul Dicks' TIHOME Club. Membership of this group will be passed to Phillip Thompson of Home and Business Microsystems in Edinburgh at the end of 1985. The group produces a quarterly newsletter, in A4 format, and is aimed primarily at the beginner and less advanced user. A disappointing aspect of their magazine is the lack of answers to readers questions sent to the Editor.

Clive Scally's group has around 1000 members and they produce the TI\*MES magazine which is produced on a regular quarterly basis. Each magazine has around 60 pages, in A5 format, of news, reviews, hints, tips, programs and articles on all TI languages. Clive has organised major user shows, the first being held in Manchester and the second in Brighton last year. This year they plan a third show in Birmingham during the third week in October. Clive also runs a business called Computer Home Service and sells mainly software. For hardware Howard Greenberg, a regular writer in TI\*MES, has a business called Arcade Hardware, 211 Horton Rd, Fallowfield, Manchester, M14 7QE. Howard always writes an amusing and informative article titled "Howard writes again", in The TI\*MES. Another enthusiast Peter Brooks writes in the same magazine and has recently formed a Regional group in Oxford the address of which is 61 The Avenue, Kennington, Oxford OX1 5PP. Peter incidentally now controls the software library of TI Home.

Another contributor to TI\*MES is Stephen Shaw who also has a software company called Stainless Steel Software, 10 Alstone Road, Stockport, Cheshire SK4 5AH. He told me that TI sold 240,000 machines in the UK. It would seem that only a small number of these have been upgraded with disk drives etc. The original purchasers had apparently been influenced to buy the computer more as a video games machine. There are even people buying up second hand machines just to get access to the TMS 9900 microprocessor at a very cheap price.

Stephen Shaw has written a book entitled "Getting Started With the TI99/4A". The book is published by Phoenix Publishing, 14 Vernon Road, Bushey, Herts.

Peter Brooks has also written a book called "Mastering The TI99/4A". Peter has a dry satirical sense of humour and you will enjoy it I'm sure. The book is published by Micro Press 27 London Road, Tunbridge Wells, Kent. Both books are to be recommended and if you want to buy a copy you should write to the publishers. The price is Stlg 5.95 or around \$12 plus postage and packing.

When I returned from the UK I found in the mail a copy of a catalogue of Software, Peripherals and Accessories for our computer published by Unisource Electronics, Inc Texas. You can obtain a copy of this catalogue by sending \$3.00 plus \$2.00 for postage to PO Box 6420 Lubbock, Texas 79462, USA. They have a number of items which are not available in Australia and they appear to be competitively priced. They also accept Visa and Mastercharge credit cards. The catalogue has over 1000 different items for our computer and they claim they ship within 48 hours of receipt of an order. So you can see our computer is still alive and kicking despite the fact that none have been manufactured for several months now. In fact we are still getting new members joining our club. I was listening to the ABC the other day and heard Les Bell, Editor of YOUR COMPUTER magazine advising his audience to join a User Group specialising in the brand of computer they had purchased, as they generally were able to give more support to the home computer owner than the manufacturer could ever achieve. This statement is certainly true for the 99/4 and 99/4A computer owner. With the reliable and advanced TMS9900 microprocessor and active User Groups around the world I think we will see our computer staying viable for a very long time.

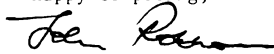
My thanks to Fred Morris for performing the secretarial duties so efficiently during my absence.

You will see that the new database program is now up and running the club's mailing list. Every member has been assigned a new membership number.

We will in future send renewal reminders each month. This will take the form of an insert in the SND.

Running out of memory.....

Happy Computing,



John Robinson.  
Hon. Secretary



5600926  
if you can  
get on that is!

MONDAY: 7PM TO 6:30 AM  
TUESDAY: 7PM TO 6:30 AM  
WEDNESDAY 7PM THROUGH TO MONDAY 6:30AM

COMING SOON: 1200/75 BAUD  
SPEED AND 300 BAUD SWITCHABLE  
We hope!

## REVIEW with KEIR WELLS



Let's put aside for a while the mindboggling games and the frustrating adventures that can be played on our ever versatile machine. There are always new games cropping up for the TI, (if you know where to look), but rarely do we come across a utility of almost unmatched power and versatility that besides filling a hitherto unfilled gap is also, and this is one of the nicer aspects, public domain. For those who don't realize what public domain software is, it's FREE.

The programme I am referring to was recently received by our club librarian, Terry, from overseas sources and is titled, 'DM-1000' (Disk Manager 1000). As can be ascertained from the title, this programme is only for those who run a disk drive with their system and also requires 32K Memory Expansion.

You may say, as I first said when I received my copy of the programme, 'So who needs another Disk Manager?', and quite rightly. The Disk Manager 2 that was put out by TI and came with the No. 1 drive has handled most tasks quite capably, albeit rather slowly, and this is one of the gaps that the DM-1000 fills.

Briefly first, the DM-1000 is a disk based Assembly Language programme. It can be either loaded using option 5 on the Editor/Assembler module or utilizing its own load programme with the Extended Basic module. When loading DM-1000 with Extended Basic the programme displays on the screen that DM-1000 is loading and that DUPLICATION IS ENCOURAGED.

Included on the disk is the documentation for the programme. There are six parts which can be printed out on either the E/A module or TI Writer. These documents are detailed and easily followed.

The programme itself is divided into three sections:

- File Utilities,
- Disk Utilities, and
- Miscellaneous Utilities.

Upon entering any of these sections it immediately becomes apparent that the DM-1000 is far superior in comparison to the Disk Manager 2.

Selecting File Utilities gives you the option to: copy files, move files, delete files, rename files, modify file protection and, best of all, recover lost files. Too many times have I heard people say that a programme that they were working on has mysteriously disappeared from their disk. This then entailed either rewriting the programme or purchasing the Navarone Disk Fixer. But no longer! The DM-1000 will retrieve and restore a 'missing' file on disk providing it has not been overwritten.

One of the beauties contained in File Utilities is that all commands within this section of the programme can be carried out without the tedious repetition required by the Disk Manager 2 module. This is made possible by a powerful screen editor which builds a catalogue of all the files and programmes on the disk and allows you to enter all the required commands at the one time. Once this has been done, the DM-1000 then takes over and executes each of the specified functions.

With Disk Utilities you are able to: catalogue disks, copy disks, sweep disks, rename disks, initialize disks and box format. While most of these commands are available with the Disk Manager 2, the speed with which the DM-1000 carries them out is brilliant. Listed below are the results of bench tests that I carried out between the DM-1000 and Disk Manager 2.

### Command DM-1000 Disk Manager 2

Initialize Disk	approx 18.6 sec's*	approx 54.5 sec's
Catalogue Disk	approx 2.8 sec's	approx 5.3 sec's
Copy 358 sec Disk	4 passes	** 9 passes

\* Initialization with DM-1000 was carried out without sector verification.

\*\*Copying a full 358 sector disk was carried out using only 1 disk drive.

With Miscellaneous Utilities the DM-1000 allows you to: install disk protection, remove disk protection, remove X/B protection, set copy disk "Bit Map" and set copy disk "Sector".

With all these commands available, DM-1000 would have to be the most powerful Disk Manager programme available for our computer.

Now that I have finished extolling the virtues of this programme let me point out something that is sorely missing...Disk Tests. I have personally found that the Disk Testing capabilities of the Disk Manager 2 to be one of its greatest attributes and to find it missing from the DM-1000 was somewhat of a disappointment. This though is the only criticism that can put forward on the DM-1000.

Credit and thanks must surely go to the author of this programme, Bruce Caron of the Ottawa TI-99/4A Users' Group.

\*\*The DM-1000 will be available from the club shop for a cost of \$5.

*Reminder!*

**MEMO**

*Our next meeting will be held on  
Saturday (2PM-4PM) September 7th  
at St. Johns Church Hall,  
Victoria Street, Darlinghurst.*

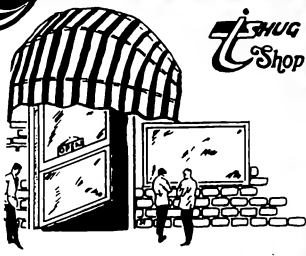
*IT'S OUR 2ND BIG*

**AUCTION  
DAY**

*Join in the fun and get a  
Bargain, or sell some of your  
gear!*

*See you there !!!*





Welcome to another Shop column.

Many members have been enquiring as to whether there is a list of tapes issued to date available. Well the good news is that there will be just such a list very shortly. I am currently putting it together and it will come complete with an order form to make it that little bit easier to order the tapes you are missing. Speaking of orders I have got a little bit behind in the mail order area lately. The reason for this is that I have had to transfer all master tapes to new tapes using the TEAC recorder. This job is now almost completed and once done there should be no problems in getting mail orders out within a week of receipt.

Those members who attended the last BBS get together will recall seeing a sneak preview of a great new program that has just arrived from our good friends in Canada. The program is called DM-1000 and it is a disk based Disk Manager with a host of new features not seen before in a program of this type. A copy has been given to our regular review writer to report on in his column so not to steal his thunder just a few words on some of the more unusual features like SWEEP DISK which enables you to re-initialise a disk in seconds, RECOVER FILE which does just that if you accidentally delete, REMOVE XB and DISK protections. I am sure you will find this an excellent utility package and the good news is it's yours for the cost of a disk (\$5). Included on the disk are full operating instructions which can be printed out with TI-WRITER or the Editor of EDITOR/ASSEMBLER.

DM-1000 will be on sale at the shop at the September meeting. You can also get it by mail by sending \$6 to the shop address.

At the August meeting several members commented on the pictures hanging on the wall. These were done with a program called PRINTDATA. As there appears quite a bit of interest in the pictures I will release the main program plus some of the picture data files on a disk also at the September meeting. The cost again will be \$5 or \$6 by mail. WARNING - the disk will contain data files which produce pictures that may be offensive to some members.

The September tape, No. 1985/9, will contain 8 programs - mainly games - most written by TISHUG members. From this issue there will be a change in rules for the lucky tape. The small program advising the lucky buyer of his/her win will be the first program on the tape. Hopefully this will ensure I get a winner each month. Still no sign of the large order placed with Navarone for Cartridge Expanders, Console Writers and DBM systems nor of Volume 5 No. 2 of HCM. With a bit of luck all will arrive by the September meeting date.

I still have plenty of copies of HCM Volume 5 Nos. 3 and 4 at \$8 each or \$9 by mail order.

Other disk software still available includes the FORTH Demo Disk, BEAXS Editor Assembler, Channel 99 programs, Enhancements for TI-WRITER & MULTIPLAN, ADVENTURE disk and the disk containing LASSO, REFORMATTER & DISKFIXER. All disk are \$5 each or \$6 each if ordering by mail.

Our subscription to MICROpendium has come through with the June and July issues just having been received. This is an excellent publication for TI Users and I recommend that you subscribe. \$35US to MICROpendium P.O. Box 1343 Round Rock Texas 78680 will get you 12 issues by airmail. The June & July issues each consist of 48 pages packed with information, reviews and new product ads. One item advertised particularly intrigued me. It's a new card for the Peripheral Box from Corcomp called TRIPLE TECH. It provides a Clock/Calendar, 64K Printer Buffer and a Speech Synthesizer Connection. With the latter you remove the speech synthesizer board from its housing and plug on to Triple Tech thus eliminating the need for the external box plugged into the side of the computer. \$109.95US will get you one of these cards from TEXCOMP P.O. Box 33064 Granada Hills CA 91344. If you want to order one I suggest you add about an extra \$20 for postage and handling.

I have just received a copy of Millers Graphics latest package ADVANCED DIAGNOSTICS. It looks quite good to me but I will give it to one of our technical members to have a look at and perhaps review. If it looks a good proposition then arrangements can be made to import and sell.

HELP. I have a program that works well on a Gemini printer but won't print properly to the Club's Epson printer. Any member familiar with control codes and printer graphics who would like to assist in re-structuring the program should get in touch with me.

On other pages are 2 programs from the library for you to type in and enjoy. The first is in Extended Basic and is called Air Traffic Controller and is a lot of fun to play. The second program is in Basic and is called Draw Poker. Hope you like them.

\*\*\*\*\*

TI.S.H.U.G C L A S S I F I E D

\*\*\*\*\*

WANT TO B U Y:

TI LOGO#2 and Double Density Disk  
Controller Card for the PHP1200  
Peripheral Expansion Box.

Phone Jim on (063)314773

WANT TO S E L L:

TI COUNT BUSINESS SOFTWARE very  
Cheap.

Phone Greg on (02)9138737

WANT TO S E L L:

64K PRINTER BUFFER as new with  
cable to connect direct to your PIO  
Printer. Ideal if you have a lot of  
typing to do, especially if you  
have a slow speed printer.

Usually \$299, will see for \$199 Or  
nearest offer.

Phone (02)5690949 after 7pm  
week-nights.

**PUBLICATION  
LIBRARY A  
REALITY**



"T I - P U B L I B"....the new TI.SHUG library service is on it's way.

Now that John Robinson has returned from his overseas sojourn I will be able to spend more time getting my library "act" together - that is, after John resumes his role once again!

Recently, I helped to tidy up Shane's place by taking over all of the exchange newsletters, cleared some of Terry's excess clutter, emptied John's cupboards of all the secreterial files AND attended to the conversion of the club data base from the PRK to the Naverone DBM system knowing, all the time, that at some time in the near future I would ultimately have to tackle the job of putting together a PUBLICATIONS LIBRARY.

Well, that time has come - a lot sooner than I would have liked. Pressure from my wife to "get the place tidied up" (she too had visited Shane's pad) now means that your library will be available to you within the next few weeks.

My thoughts as to the manner in which the library business should be conducted have yet to be confirmed by the Co-Ordinating Committee - this I hope to have done at the August meeting of the committee.

As I see it, the library will offer the following facilities:-

- A magazine library containing any and all magazines purchased by and/or donated to the club. Fully indexed and where possible volumes containing successive editions of a particular magazine.
- A book library containing publications devoted to the TI and which deal with Programming languages, Technical aspects, Education and Applications.
- A module library containing any TI and /or third party cartridge based software - all donated by the members!
- A newsletter library made up of all the exchange publications received by TI.SHUG.
- Information bulletins prepared from all sources dealing with specific aspects of programming languages, technical "adventuring", and applications. In this respect Chris Buttner has already started on putting together 2 information bulletins - one dealing with the PRK enhanced TI. Basic and the other a series of articles dealing with the Disk and how the TI. organises the directory and files. (this info. is ideal for HACKERS!)

As you can imagine, the task is fairly awesome! You can help! This is how you can assist me to start the ball rolling. I need you to donate any publication, be it a magazine, book, module, technical manual and/or instruction booklet that you think may be of interest to others. (the fact you have even got it to donate means that it MUST be of some TI significance!). Ofcourse, you would also have finished with it and can now pass it on to the library!

A plea! If anyone out there has knowledge about library maintenance/management I would like to hear from you. The more professional the initial setup then the better the end result. (think so?)

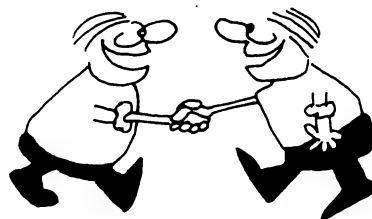
You may do this by either mailing your donation/contribution to me c/o TI.PUBLIB. P.O. Box 149, PENNANT HILLS 2120. NSW. or hand it to any one of the committee members at the club meeting.

## CURRENT LIBRARY HOLDINGS

- A FEW HUNDRED YET TO BE CATALOGUED EXCHANGE MAGAZINES.
- A NUMBER OF BOOKS YET TO BE CATALOGUED.
- A FEW TECHNICAL MANUALS.
- A COUPLE OF INFORMATION BULLETINS (Well, thats what they seem to be).
- AND ME!

See you next issue with some more positive feed back. Oh, incidentally, I was talking to Ross Mudie the other day - I forget the topic. However, I was making an arrangement to contact him at work when he said to me that his best "WINDOW" time was between 7:30am and 2:30pm. What strikes me as significant is how the English language changes to suit the current "in thing". WINDOWS ARE IN are they not? (Sorry Ross, I couldn't resist it!)

Regards, Fred Morris.



**Get Acquainted!**  
at



\*\*\*\*\*

## APOLOGIES

For all those avid readers of Techo Time I regret to inform you that due to the size of this issue of the SND and the size of Roberts' article, Techo Time has been put back for the next issue. I apologize to those who were waiting for Roberts article but I can assure you that the wait will prove more than worth while.

SHANE.

\*\*\*\*\*



## A Letter

S  
H F C L O  
P A ! a N

This letter was received from one of our members who lives in Victoria.

Dear SND,

Being my first communication with you since joining some months ago please forgive the length of this letter but as you will realise, my locality poses a problem to frequent contact.

I will endeavour to keep some order to my queries as it's possible they might require contact with more than one member.

Is it possible to have back copies sent down to me -ie. prior to Nov 84. If so indicate cost.

One suggestion: Could it be possible to publish a list of the items that are to be ordered from overseas with a deadline so that members on seeing what was ordered could be involved with the purchase also.

To explain further I realise this may be done to some extent but for someone like me, I never seem to understand what a lot of those articles actually do.

Which Disk Fixer do you recommend and what does it actually do?

I have the TI Expansion Box and Memory Card and DS/DD drive.

I assume the Corcomp cards don't fit - is this right?

The best way to go modem taking into account the disk versions of the E/A and TEII?

I would like to quieten my fan but am confused over the two opposing articles viz. the one to fix overheating and the cutting down of fan speed.

If no problem could a pot be used instead of a fixed resistor? If so could this be mailed down to me as I am unable to find one here?

In the future I intend to transfer my computer on board a sailboat. Is it possible to convert a TI system to 12 volt.

viz. Are there power supply systems to enable eg. Expansion System, printer, modem & RTTY all from 12 volt.

As I am interested in adventures could you tell me the price of the Infocom ones and if you bulk buy for members if they send in the money.

Enclosed is a Data compression that I found is useful for saving memory in text adventures that have much text that is repetitive. It may or may not be used already but it does help people without the Expansion Card to write more complete stories. I usually use it to describe locations. - it would of course have other applications.

I got the idea from an English magazine.

Again, sorry about the length and thanks to the committee for keeping alive our TI's.

All the best in the future,

Barry Ridgeway.

PS. I would like to see the Communicators section continued as this appears a vital part of our hobby and is a growth area for the future.

\*\*An idea for Data Compression.

```
100 DIM C$(25)
110 C$(1)="THE FIRST REPETITIVE SENTENCE"!ENTER C$(1)
AS /A
120 C$(2)="THE SECOND REPETITIVE SENTENCE"!ENTER C$(2)
AS /B
130 C$(3)="ETC. WHATEVER YOU WANT TO FIT IN BETWEEN THE
INVERTED COMMAS - RIGHT UP TO 25 IN NUMBER"
140 READ D$
150 FOR I=1 TO LEN(D$)
160 IF SEG$(D$,I,1)<>"/" THEN PRINT SEG$(D$,I,1); ::
GOTO 190
170 PRINT " ";
180 C=ASC(SEG$(D$,I+1,1))-64 :: PRINT C$(C); :: I=I+1
:: PRINT " ";
190 NEXT I
200 DATA THIS IS/AHOW/BIT/CWORKS
```

First of all Barry thank you very much for the Data Compression programme and I am sure that other members of the club will find it interesting and useful.

In regards to your query about the Infocom series of adventures, I apologise to one and all for neglecting to include ordering information in my Review Column on these programmes. For all those who wish to order the Infocom adventures, you can obtain them from:

TENEX COMPUTER EXPRESS.  
PO BOX 6578  
SOUTH BEND, IN 46660

Unfortunately the club library does not order or bulk buy from TENEX. You might also like to mention to them that you are a member of TISHUG and ask them for a catalogue of their 99/4a products.

For back copies of the SND you will need to request these from the club secretary John Robinson and enclose \$2 for each issue required.

For availability of items from overseas refer to the shop column in the SND for an up to date list of items that are currently available. If there is something that you wish to purchase and does not appear currently in the shop column then write to the club librarian, Terry Phillips.

In regards to the Corcomp cards, Corcomp do make cards that do fit in the PE Box. The cards that they do produce are: DS/DD Disk controller card, 32K card and RS232 card (both these have been produced but I am not aware of their availability in Australia). It might be advisable to contact Imagic Australia as to their current status with these items.

Disk Fixers will be reviewed in the next issue of Review Column.

Unfortunately I am not able to answer your question about modem communications, fan quietening and conversions of power supplies but I have passed on these queries to the respective members of the committee and the answers will be published in the next issue of the SND.

Once again thanks for the letter and good wishes.

Regards,

Keir Wells (KEIRLALOR)

A new column will be added to my REVIEW COLUMN for general queries regarding software and such items. If you have any questions then please write to me care of the club address. (Do not include queries on specific items that are catered for by other columnists eg. Techo time).





## MY FIRST PROGRAMME.

### 1.0 MOTIVATION.

I saw a BASIC programme in MICROSOFT for a calendar. It had a wider range than mine. I couldn't understand the algorithms the author used so I couldn't translate it for the TI.

I set about building my own CALENDARS programme, using WORLD BOOK Encyclopedia as my reference.

### 2.0 OUR CALENDAR.

We use the GREGORIAN calendar which was proposed by Pope Gregory XIII in the 1580's. It was not until 1753 that Britain adopted this calendar.

It started on Monday, 1st. January, 1753.

### 3.0 EXTRA DAYS.

To keep the calendar in line with the seasons, it was necessary to add an extra day each four years. Leap years are well known for their 29 day Februaries. Not so well known, is that every century except for those years divisible by 400, February has only 28 days as in a normal year.

2000 a.d. will be such a year. If I want to see a leap year with only 28 days in February, then I will have to stick around for another one hundred and fifteen years!

### 4.0 MY ALGORITHMS.

Look at the programme listing. I have left the tail remarks intact so that you might begin to appreciate the tortuous experience I had, as a new programmer, writing my programme by thumbing through the BASIC book "USER'S REFERENCE GUIDE".

After many attempts with TI BASIC, I abandoned hope of ever getting the TABS to co-operate with me.

The XB was much more helpful and I began to learn how to set up arrays and use Data Statements. The logical expressions for determining whether we had a leap-year or not were real doozies at first, till I understood how to express them. Look closely at the expressions for adjusting the count of days start and the TAB for printing according to the logic of Leap Year?, Century Year?, 4th Century Year? etc.

Well, I hope you like my expose' of at least 400 hours work, and hope it may give somebody else an insight into how to construct a programme.

Alan Swales.  
3/8/85

\*\*Alans' programme appears in the programme section of this issue of the SND.

## EVERYTHING AND THE MINIMEM.

From TI-TIMES by Stephen Shaw.

In the previous two editions of the SND we reprinted the following:-

- SPEECH AND THE MINIMEM
- SOUND AND THE MINIMEM

Now Stephen asks:-

"Remember SPRITES, magic things, you merely tell the computer to start one moving and off it goes no matter what else the computer is doing. It just keeps on moving? Wouldn't it be nice to have music do that? Give the computer the score, tell it to start playing and then get on and do something else?"

I hope you are sitting down because it may come as a shock to find out that the TI 99/4A can actually do 3 things at once.

1. Move a SPRITE
2. Play a musical piece
3. RUN a program.

The catch is that you need to place the music score somewhere...and that somewhere is in VDP RAM. Any spare space tends to be pushed into by the variable garbage.

Fortunately it is possible to reserve VDP RAM and the following program will amaze you. It comes from Neil Lawson again, who in the SMART PROGRAMMER was using machine code to give him access to VDP using Extended Basic.

I can't reprint that program so I have amended it to work with the MINIMEM.

To set up the program:-

Turn your console on and select TI Basic.  
If you have a disk controller, you can use:  
CALL FILES(8)  
then NEW  
then key in the program and run it.

If you do not have a disk controller, you can still reserve memory. The CALL FILES reserves space at the top of VDP RAM and this is controlled by a couple of bytes in CPU RAM... we can use CALL LOAD(-31888,50,0) to do the same thing.

Type in CALL LOAD(-31888,50,0) then NEW.  
Then key in the program and run it.

The NEW is important! It tell the computer to remap VDP. This it does after reference to the location -31888.

Loading -31888 with 50 and 0 tells the computer that the TI Basic program can commence not at it's normal 16383 but instead at 256\*50+0=12800.

This gives us about 3.5K for the music, far more that the program needs!

The program now, then the ENLIGHTENMENT. The CALL SOUND in line 100 is essential so don't miss it out!

```

100 CALL SOUND(-2,30000,0)
110 CALL CLEAR
120 REM DURATION IN 1/60TH
130 REM OF A SEC.1 SECOND=60

140 REM 1/2 SECOND=30
150 REM *****
160 REM
170 REM BELOW LOADS SOUND TA
BLE
180 CALL POKEV(14096,3,142,1
5,144,30)
190 CALL POKEV(14101,3,133,1
3,144,60)
200 CALL POKEV(14106,3,128,1
5,146,30)
210 CALL POKEV(14111,3,142,1
5,144,15)
220 CALL POKEV(14116,3,141,1
7,145,15)
230 CALL POKEV(14121,3,142,1
5,146,30)
240 CALL POKEV(14126,3,129,2
0,146,30)
250 CALL POKEV(14131,3,141,1
7,144,15)
260 CALL POKEV(14136,3,129,2
0,146,15)
270 CALL POKEV(14141,3,131,2
1,146,30)
280 CALL POKEV(14146,3,140,2
3,146,30)
290 CALL POKEV(14151,3,139,2
6,144,60)
300 CALL POKEV(14156,3,141,1
7,146,30)
310 CALL POKEV(14161,3,141,1
7,144,60)
320 CALL POKEV(14166,3,142,1
5,146,30)
330 CALL POKEV(14171,3,133,1
3,144,60)

340 CALL POKEV(14176,3,128,1
5,146,30)
350 CALL POKEV(14181,3,142,1
5,144,15)
360 CALL POKEV(14186,3,141,1
7,145,15)
370 CALL POKEV(14191,3,142,1
5,146,30)
380 CALL POKEV(14196,3,129,2
0,146,30)
390 CALL POKEV(14201,3,141,1
7,144,15)
400 CALL POKEV(14206,3,129,2
0,145,15)
410 CALL POKEV(14211,3,131,2
1,146,30)
420 CALL POKEV(14216,3,140,2
3,146,30)
430 CALL POKEV(14221,3,129,2
0,144,60)
440 CALL POKEV(14226,3,134,0
0,159,30)
450 CALL LOAD(-31796,55,16)
460 CALL LOAD(-31794,1)
470 REM
480 REM
490 REM THIS LOOP WILL BE
500 REM PROCESSED WHILE THE
510 REM MUSIC PLAYS
520 REM TWO PROGRAMS
530 REM AT ONCE!
540 REM -----
550 CALL CLEAR
560 PRINT " ]": " ]
]": " ]]]": " : : : :
570 CALL PEEKV(1152,A,B,C,D,
E,F,G,H)
580 CALL POKEV(1232,A,B,C,D,
E,F,G,H)
590 REM CHR$(58) DEF AS ZERO

```

```

600 REM
610 FOR T=1152 TO 1231
620 CALL PEEKV(T,A,B,C,D,E,F,G,H)
630 CALL POKEV(1512,A,B,C,D,E,F,G,H)
640 CALL PEEK(-31796,A,B)
650 IF A<55 THEN 690
660 IF B<150 THEN 670 ELSE 690
670 NEXT T
680 GOTO 610
690 REM

700 REM END OF LOOP
710 CALL SOUND(-20,30000,0)
720 REM
730 REM "READY FOR NEXT RUN"

740 REM WHEN SOUND TABLE DONE
750 J=J+1
760 IF J<40 THEN 750
770 J=0
780 CALL LOAD(-31796,55,16)
790 CALL LOAD(-31794,1)
800 GOTO 610

```

LINE 100 is required for the MINIMEM. For the Extended Basic version Neil had to use:

```

455 CALL PEEK(-31747,A)::CAL
L LOAD(-31747,1 OR A)

```

That a logical 'OR'

Neil's program used CALL LINK("POKEV"....) to access VDP via a machine code utility.

The music is loaded next and you will see the area of VDP memory used is within the reserved area.

The first item of the POKEV is the address the first byte is going to be loaded at. The sound table has to be sequential!

The we load that and successive locations as follows:

1. Length of sound information, excluding duration, normally 3.
- 2 & 3. The tone generator data, exactly as in the previous direct sound programs, as yielded by Neil's Utility programs earlier.
4. Attenuation (or volume) value, again as in direct sound.
5. The time the note is to sound, is in 60th's of a second.

LINE 550 tells the computer where the SOUND TABLE can be found: 55\*256+16=14096. (Check line 280!)

LINE 580 is an instruction to the computer to start playing.

Then we go into a loop which demonstrates that the computer carries on playing without further instruction.

PROBLEM: How do we know the music has ended? The computer only does as it is told, and will carry on through memory until you tell it to stop... corrupting the program as it goes along.

My attempt at this was to wait till the end of the tune, BREAK and find out the value at -31796 and -31795. Then I test for those values, as above.

LINE 770 restores sound to normal and LINES 750 to 770 put a small delay before we start again by loading the start of the sound table and tell the computer to play!

Neil had a different idea... place a silent note (eg. 30db attenuation) at the end of the sound table. Test another location:

```

Add:
545 CALL POKEV(-14231,3,159,
191,223,0)

```

Then instead of 640 to 660 use:

```

650 CALL PEEK(-31747,A)
660 IF A=0 THEN 780

```

I hit a problem translating this for MINIMEM, it did not work! The solution was to reintroduce that dropped location -31747, by adding to the above program in place of 750 to 770:-

```

750 CALL LOAD(-31747,1)

```

Now for some experimenting.... Get the sound table going and before it has finished try:-

1. Using CALL SOUND with a positive time value.....
2. Using CALL SOUND with a negative time value..... using direct sound access.

OK? That's your work for the day. If you don't already have a MINIMEM here is your excuse to go out and buy one.

My work was cutting the sound length down to 1/60th of a second for better envelope shaping...and note you can change frequency as fast as the volume, for interesting glides.

Oh, if you wish to have a disk system and wish to load the program from disk, a tiny problem occurs: You have to move the VDP floor before loading from disk, and depending on the value you load it could lock up the disk system. Save the program to EXPMEM2 as shown earlier, then move the VDP floor!

THIS ARTICLE CONCLUDES THE 3 PART SERIES. STEPHEN SHAW HAS INVITED COMMENTS AND/OR QUERIES REGARDING THE ABOVE TO BE DIRECTED TO HIM AT:-

10 ALSTONE ROAD  
STOCKPORT  
CHESHIRE. SK4 5AH. UNITED KINGDOM.

For a direct reply a stamped self-addressed envelope is essential. It could take a couple of weeks to get back to you. -ED.

```

100 REM CANNEL 99 HAMILTON USER'S GROUP
110 REM TEAM PICKER
120 REM EXTENDED BASIC
130 REM *****
140 REM * BY MIKE TOWERS *
150 REM *****
160 GOTO 240
170 CALL CLEAR
180 CO=(28-LEN(T$))/2
190 DISPLAY AT(1,CO):T$ :: DISPLAY AT(4,1):"WINS:" :: DISPLAY AT(6,1):"LOSES:" ::
: DISPLAY AT(8,1):"TIES:"
200 DISPLAY AT(10,1):"POINTS FOR:" :: DISPLAY AT(12,1):"POINTS AGAINST:"
210 ACCEPT AT(4,8)VALIDATE(DIGIT)SIZE(2):WINS(P):: ACCEPT AT(6,9)VALIDATE(DIGIT)
SIZE(2):LOSES(P):: ACCEPT AT(8,8)VALIDATE(DIGIT)SIZE(2):TIES(P)
220 ACCEPT AT(10,14)VALIDATE(DIGIT)SIZE(4):F(P):: ACCEPT AT(12,18)VALIDATE(DIGIT)
)SIZE(4):A(P):: PCT(P)=(WINS(P)+(TIES(P)*.5))/(WINS(P)+LOSES(P)+TIES(P))
230 RETURN
240 CALL CLEAR
250 INPUT "HOW MANY GAMES? ":GAMES
260 FOR LOOP=1 TO GAMES
270 INPUT "VISITORS ":VIS$
280 PRINT :: INPUT "HOME ":HOME$
290 VIS=1 :: P=VIS :: T$=VIS$ :: GOSUB 170
300 HOME=2 :: P=HOME :: T$=HOME$ :: GOSUB 170
310 IF PCT(VIS)=PCT(HOME)THEN 350
320 IF PCT(VIS)>PCT(HOME)THEN WINNER$=VIS$ ELSE WINNER$=HOME$
330 CALL CLEAR :: PRINT "THE WINNER WILL BE:";WINNER$ :: NEXT LOOP
340 END
350 IF WINS(VIS)=WINS(HOME)THEN 380
360 IF WINS(VIS)>WINS(HOME)THEN WINNER$=VIS$ ELSE WINNER$=HOME$
370 GOTO 330
380 IF F(VIS)=F(HOME)=F(HOME)-A(HOME)THEN WINNER$=HOME$ ELSE 400
390 GOTO 330
400 IF F(VIS)=F(HOME)>F(HOME)-A(HOME)THEN WINNER$=VIS$ ELSE WINNER$=HOME$
410 GOTO 330
420 END

```

**END**

## REGIONAL REPORT: Reports from our Regional Home-group leaders...

"See you there 'cause we care"

### LIVERPOOL REGIONAL MEETING

DATE: 9th AUGUST, 1985.  
PLACE: CAMPBELLTOWN  
HOST: STAN PUCKLE

We had a roll up of 14 members from as far a field as Northmead and The Oaks.

A summary of the TISHUG meeting at the cross was given by Robert Peverall.

We note with interest the forthcoming full day tutorial to be held at Burwood.

We discussed a number of items and list the two that may be of the most interest,

1. It was thought that possibly a raffle could be organised thru the main body, it was thought that a Corcomp system as 1st prize, with maybe a disc controller for 2nd might attract a lot of interest.

2. We propose a family picnic day either late November or early December at Picnic Point. We extend an invitation to all fellow users to attend. Those parties interested, would you please contact any of the following persons and advise prospective numbers.

Ross Hardy: 6376772  
Stan Puckle: 046-256157  
Hans Zecevic: 600-8716  
Cyril Bohlsen: 639-5847

We continued with a refresher of last months tutorial on E/A and Debug and then continued with with Part 2.

We once again extended our thanks to Kevin Gardner for an excellent presentation, and effort in research and preparation of documentation.

We held demonstrations of various new pieces of new software and the inevitable backup.

### FORTHING MEETINGS

SEPTEMBER 14TH: ROSS HARDY 15 EXCELSIOR  
GRANVILLE

OCTOBER 11TH: CYRIL BOHLSSEN 4 MADELINE  
NORRITMEAD.

NOVEMBER 13TH: MARCEL 35 FEARN ST.  
631-1534 TOONGABBIE

Until next month happy byting.

Since I moved from Sydney to Deniliquin about 4 months ago, I have been able to make contact with two other TISHUG members living here. The following is a report on our first Regional Meeting for the next issue of SND.

### CENTRAL MURRAY REGIONAL GROUP. \*\*\*\*\*

The first meeting of this new regional group was held at Deniliquin in July. Deniliquin is a country town about 700km Southwest of Sydney and 200km West of Albury. Out of a population of 8,000, three TISHUG members and their families met on a Sunday afternoon. The expanded system was demonstrated with particular interest being shown in the use of the printer. TI-Writer was looked at briefly as well as GRAPHX and other software, such as the Physical Fitness module. The meeting concluded with the sharing of some programmes.

The next meeting will be held in September and anyone living in the region is invited. Please contact:

Ray Brown  
(058)81-2360

Some time will be spent at the next meeting looking at members' programmes and de-bugging them.

\*\*\*\*\*  
\* \*  
\* BANKSTOWN REGIONAL GROUP \*  
\* \*  
\*\*\*\*\*

Our last meeting was held at David Mayo's place and again didn't finish till almost midnight. Shane Ferret helped one of our younger members to debug a filing program. This same lad, Mathew demonstrated a light-pen he had constructed. Well done Mathew!

The whole evening was a question and answer session on a wide variety of topics with everyone sharing and gaining a lot.

Meetings are held on the third Sunday of each month starting at 7:00PM. For more information phone David on (02)708 4293 or leave a message on the BBS for MAYDAY.

### CENTRAL COAST REGIONAL GROUP.

We wish to notify all members of TISHUG, that the Central Coast Group will be conducting a full day workshop. All members are invited to attend the workshop with their Computers, to enjoy the company of other members and exchange knowledge of their programming.

DATE: -Saturday 19th October

TIME: -9:30AM to 4:00PM. Snack served for lunch. PLACE: -Gorokan High School  
Goobarabah Ave, (of Dudley St.) Gorokan 2263.  
Enquiries: - Russell Welham 043 924000

### WORKSHOP WORKSHOP WORKSHOP WORKSHOP WORKSHOP

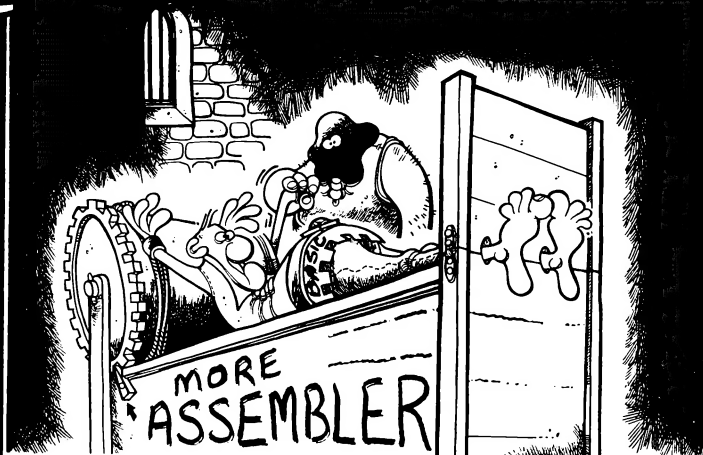
A L L W E L C O M E !

At the meeting held in August, I showed how to clean the computer. Program demos and questions were answered.

### WARNING WARNING WARNING

When pulling the computer apart be careful of the connection between the keyboard and the main board, the lead plugs into the main board, Remove the plug don't bend the main board back too many times as I just had to rewire a computer that this connection had broken.

Russell.



```

*****
* CHANEL 99 HAMILTON USER'S GROUP *
* PICTURE SPELL by D. STOREY *
* Translated to TMS9900 machine code *
* by Iain Johnson *
*****
DEF START program entry point
REF VMBW,VMBR labels for routines in low
REF VSWB,VSWR memory
REF VWTR,KSCAN,GPLINK,XMLINK
STATUS EQU >837C equate the following labels
KEYNUM EQU >8374 to constants used during
KEYVAL EQU >8375 assembly of this program
FAC EQU >834A when the FAC label is found
SPACE EQU >2020 by the assembler it will
MYREG EQU >8300 replace it with >834A
FUNCTS BYTE 8 initialize bytes to values
FUNCID BYTE 9 pointed to by these labels
ENIKEY BYTE 13 i.e. ENIKEY will point to
YES BYTE 89 the value 13 (ascii enter)
NO BYTE 78 NO points to value ascii 78
EVEN make even word boundary
TEXTBUF BSS 16 buffer will hold input word
MASK DATA >2000 status equal bit check mask
SAVRET DATA 0 buffer to save return addr.
HEX2 DATA 2 value 2 called by label HEX2
POINT DATA 0 buffer to record score in
* TEXT FOR MESSAGES
MSG$1 DATA >0069,14 first word = addr on screen
TEXT 'PICTURE SPELL '
MSG$2 DATA >00AA,12 second word = # bytes
TEXT 'By D. STOREY'
MSG$3 DATA >00C5,22 address, number of bytes
TEXT 'translated to TMS9900 '
MSG$4 DATA >00E9,12 address, number of bytes
TEXT 'machine code'
MSG$5 DATA >0108,16 address, number of bytes
TEXT 'By Iain Johnson '
MSG$12 DATA >0269,14 address, number of bytes
TEXT 'PRESS ANY KEY '
MSG$6 DATA >02C6,18
TEXT 'SPELL THIS PICTURE'
MSG$7 DATA >0267,16
TEXT 'THAT IS CORRECT '
MSG$8 DATA >0267,16
TEXT 'PLAY AGAIN Y/N? '
MSG$9 DATA >267,18
TEXT 'THAT IS INCORRECT '
MSG$10 DATA >2E2,14
TEXT 'YOUR SCORE IS '
MSG$11 TEXT 'OUT OF'
EVEN
* TEXT FOR CHECKING IF INPUT IS CORRECT
TAPPLE BYTE 5 length of string
TEXT 'APPLE' word string = apple
TDOOR BYTE 4 length of string
TEXT 'DOOR' word string = door
TCAR BYTE 3
TEXT 'CAR'
TSHIP BYTE 4
TEXT 'SHIP'
THOUSE BYTE 5
TEXT 'HOUSE'
EVEN

```

```

*****
* PROGRAM BEGINS HERE *
*****
* MAIN PROGRAM MODULE *
*****
START MOV R11,@SAVRET allows return to menu
LWPI MYREG load my workspace registers
LIMI 0 disable vdp interrupts
BL @CLEARS clear screen
LI R0,>08F8 char def table for edge chr
LI R1,BLOCK solid block definition
LI R2,8 8 bytes in character def
BLWP @VMBW define edge char as a block
CLR R6 R6 = TURN = 0
CLR R8 uses less MEMORY than LI R8,0
CLR @POINT variable POINT=0
BL @HCHAR branch to hchar subroutine
DATA >385,>E600,11 using data directives
to load the
address,char,# repetitions
LI R0,>383 colour table for edge char
LI R1,>E600 colour gray on gray in msb
BLWP @VSWB colour char set 3 gray/gray
INC R0 addr in colour tab of space
LI R1,>6600 dark red on dark red
BLWP @VSWB write color byte to vdp ram
LI R0,>070E value of >E for reg 7
BLWP @WTR call screen colour >E=GRAY
* MAKE BORDER
BL @VCHAR multiple vchar statements
DATA 2 perform 2 vchar statements
DATA >0000,>1F00,48 addr,char,repetitions
DATA >001E,>1F00,48 addr,char,repetitions
*****
* DISPLAY TITLE SCREEN *
*****
BLWP @DISPLA gosub to routine to print
DATA MSG$1 text on screen letter by
letter
DATA MSG$2
DATA MSG$3
BLWP @DISPLA
DATA MSG$4
BLWP @DISPLA
DATA MSG$5
BLWP @DISPLA
DATA MSG$12
CLKE5 BL @CALLKE branch to keyboard scan
COC @MASK,R0 key pressed?
JNE CLKE5 if not, scan keyboard again
* CLEAR ROW 19 WITH SPACES
BL @HCHAR branch to hchar using data
DATA >262,SPACE,28 addr,char,repetitions
* PUT SPACES IN 3 ROWS
START2 BL @HCHAR multiple hchar's using data
DATA 3 3 hchar statements
DATA >267,SPACE,15 spaces in row 19
DATA >2C6,SPACE,18 spaces in row 22
DATA >162,SPACE,10 spaces in row 11
START3 BLWP @DISPLA display "spell this picture"
DATA MSG$6
INC R6 turn = turn + 1
* PICK A NUMBER FROM 0 TO 4

```

```

RNDMOR BL @RANDOM branch & link(gosub) random
SRL R1,13 shift right 13 bits
CI R1,4 compare register 1 to val 4
JGT RNDMOR if greater than 4 pick again
MPY @HEX2,R1 else R2 = R1 * 2
LI R1,VECTOR R1 = addr of branch address
A R1,R2 add random# * 2 to vector
MOV *R2,R1 move branch addr to R1
BL *R1 equivalent to on gosub
JMP AFTVEC skip over vector data
VECTOR DATA APPLE,DOOR,CAR,SHIP,HOUSE branch addr
AFTVEC BL @HCHAR call hchar sub & pass data
DATA >0162,SPACE,15 to subr using reg 11
BLWP @INPUT call input routine
* COMPARE INPUT STRING WITH ACTUAL WORD e.g. DOOR
LI R3,TEXTBUF reg 3 points to input text
MOV R9,R0 reg 9 points to text word
MOVB *R0,R2 move length byte to R2 msb
SRL R2,8 make valu in R2 a word valu
CHKMOR CB *R3+,*R0+ compare text char by char
JNE WRONG if text not match jumpWRONG
DEC R2 next ascii char in string
JNE CHKMOR finished?
BLWP @DISPLA display "that is correct"
DATA MSG$7
* BLANK OUT PICTURE SQUARE
LI R9,>133 square starts at row9 col19
LI R4,5 five columns to blank out
VCHMOR MOV R9,R0 start to blank square
LI R1,SPACE R1 loaded with >2020 spaces
LI R2,8 8 repetitions of space
BL @VCHAR vchar routine using R0,R1,R2
INC R9 next column over to right
DEC R4 decrease column counter by 1
JNE VCHMOR finished? if not,more vchar
BL @HCHAR blank out "that is correct"
DATA >267,SPACE,15 15 spaces in row 19
INC @POINT score increases by 1
BLWP @DISPLA display "play again y/n" by
passing address of message
DATA MSG$8
YNASK BL @CALLKE branch to call key routine
COC @MASK,R0 compare status in R0 to mask
JNE YNASK if not equal call key again
CB R1,@YES else compare ascii to "Y"
JNE IFNO if not "Y" go check for "N"
B @START2 if "Y" pressed play again
IFNO CB R1,@NO compare input key with "N"
JNE YNASK if not equal,call key again
* FINISHED PLAYING GAME
BL @CLEARS clear screen with spaces>20
BLWP @DISPLA display at "your score is "
DATA MSG$10 address of text
MOV @POINT,@FAC put score at fac to
BL @INTSTR convert score to string
LI R0,>2EF row 23 column 15
BLWP @VMBW print score in row 23
A R2,R0 add length of score string
INC R0 to screen print position
LI R1,MSG$11
LI R2,6
BLWP @VMBW display "out of"
AI R0,6 adjust screen print position
MOV R6,@FAC mov # turns to f.a.c.
BL @INTSTR convert # turns to string
BLWP @VMBW print string of # turns
COC @MASK,R0 delay until key pressed
JNE CLKE7
RETURN MOV @SAVRET,R11 return to selection
MOV R8,@STATUS screen
RT
* INCORRECT ANSWER
WRONG DEC @POINT score>> point = point + 1
JLT WRONG2 if point less than 0 WRONG2
JMP WRONG3 else jump WRONG3
WRONG2 CLR @POINT point = 0
JMP YNASK jump to "play again y/n"
WRONG3 LI R4,2 2 times to display message
* DISPLAY 'THAT IS INCORRECT' TWICE
WRONG4 BLWP @DISPLA
DATA MSG$9
BL @HCHAR erase message"that is inco"
DATA >267,SPACE,17
DEC R4
JNE WRONG4 finished displaying message?
B @AFTVEC go and pick another picture
** END OF MAIN CONTROL PROGRAM **

```



\*\*\*\*\*  
\* SUBROUTINES \*

\*\*\*\*\*  
\* DISPLAY TEXT SUBROUTINE \*

\*\*\*\*\*  
DISPLA DATA SUBWS,\$+2 workspace,jump to next line  
MOV \*R14+,R4 R4= address of first string  
MORMES MOV \*R4+,R0 mov address from data to R0  
MOV \*R4+,R2 R2=number of bytes to write  
MORTX MOV \*R4+,R1 R1= character to write(msb)  
BLWP @VSW write first letter to screen  
INC R0 next screen position  
CB R1,SPACE if letter is a space  
JEQ SKIP then skip delay loop  
LI R5,>1800 delay loop  
DELAY1 DEC R5  
JNE DELAY1  
SKIP DEC R2 finished typing to screen ?  
JNE MORTX if not write another letter  
RTWP else return to calling sub  
\*\*\*\*\*

\*\*\*\*\*  
\* INPUT SUBROUTINE \*

\*\*\*\*\*  
INPUT DATA SUBWS,\$+2 workspace for sub is SUBWS  
\*\* CLEAR TEXT BUFFER(in memory at address TEXBUF)  
LI R4,TEXBUF R4 points to input buffer  
LI R2,16 16 bytes to load blanks to  
LOOP3 MOV @SPACE,\*R4+ clear buf word by word  
DECT R2 R2 = R2 - 2  
JNE LOOP3 if not finished, blank more  
\*\* SET VARIABLES FOR KEYBOARD INPUT SUBROUTINE  
LI R3,>162 left margin for cursor  
LI R4,>170 right margin for cursor  
LI R5,>5F00 ascii code for cursor ' '  
LI R7,300 cursor blink on rate =300  
LI R8,150 cursor blink off rate =150  
LI R12,>162 temporary cursor position  
\*\*\*\*\*

\*\*\*\*\*  
\*\* GET ASCII CODE OF CHAR UNDER CURSOR IN REG 6  
CURGET MOV R12,R0 load R0 with cursor position  
BLWP @VSR read character on screen R1  
MOV R1,R6 left byte R6 holds character  
\*\* PUT CURSOR ON SCREEN  
CURPUT MOV R12,R0 mov cursor position to R0  
MOV R5,R1 mov cursor ascii code to R1  
BLWP @VSW put cursor on screen  
\*\*\*\*\*

\*\*\*\*\*  
\*\* CURSOR FLASH ON LOOP  
MOV R7,R9 load R9 with loop delay len  
LOOP4 BL @CALLKE branch to keyboard subroun  
COC @MASK,R0 test status  
JEQ GOTVAL if status equal goto GOTVAL  
DEC R9 else decrease delay length  
JNE LOOP4 if loop < 0 then call key  
MOV R12,R0 load cursor position to R0  
MOV R6,R1 R1=ascii code cursor  
BLWP @VSW put cursor on screen  
\*\*\*\*\*

\*\*\*\*\*  
\*\* CURSOR FLASH OFF LOOP  
MOV R8,R9 load R9 with loop delay len  
LOOP6 BL @CALLKE same as cursr flash on loop  
COC @MASK,R0  
JEQ GOTVAL  
DEC R9  
JNE LOOP6  
JMP CURPUT no key presdStart loop agen  
\*\*\*\*\*

\*\*\*\*\*  
\*\* A KEY HAS BEEN PRESSED  
GOTVAL CB R1,@ENTKEY compare key with enter key  
JEQ ENTER jump if equal to enter loop  
CB R1,@FUNCS compare key with backspace  
JEQ BACKUP jump if equal to backspace  
CB R1,@FUNCTD compare key to function d  
JEQ FORWARD jump if equal to forwardspc  
MOV R12,R0 load R0 with screen position  
BLWP @VSW write input key char on scr  
INC R12 next screen position  
C R12,R4 comp cursr pos to rite marg  
JEQ ENTER if equal enter value  
JMP CURGET else start cursor loop  
\*\*\*\*\*

\*\*\*\*\*  
\*\* ENTER KEY PRESSED  
ENTER CB @MASK,R6 compr the char under cursor  
JEQ ENTY to space- if equal go ENTY  
MOV R6,R1 else load R1 to print saved  
JMP ENTX character  
ENTY LI R1,SPACE load R1 to eliminate cursor  
ENTX MOV R12,R0 load cursor screen position  
BLWP @VSW put spc or charR6 on cursor  
LI R0,>162 load R0 with text on screen  
LI R1,TEXBUF load R1 with text buffer  
\*\*\*\*\*

\*\*\*\*\*  
\*\* FUNCTION S  
BACKUP C R3,R12 if cursor at margin skip  
JEQ CURGET this routine  
MOV R6,R1 load R1 with saved char  
MOV R12,R0 load R0 with screen position  
BLWP @VSW write saved char on screen  
DEC R12 cursor.pos = cursor.pos - 1  
JMP CURGET go to beginning cursor loop  
\*\*\*\*\*

\*\*\*\*\*  
\*\* FUNCTION D  
FORWARD C R4,R12 if cursor st margin skip  
JEQ CURGET this routine  
MOV R6,R1 load R1 with saved char  
MOV R12,R0 load R0 with screen position  
BLWP @VSW write saved char on screen  
INC R12 cursor.pos = cursor.pos + 1  
JMP CURGET go to beginning cursor loop  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\* returns ascii code of key  
\* CALL KEY SUBROUTINE \* pressed in left byte R1  
\*\*\*\*\* and status in R0 msb  
CALLKE MOV R8,@STATUS clear status byte  
LI R2,>0300 key code number = 3  
MOV R2,@KEYNUM mov key code to >8374  
LIMI 0 disable v d p interrupts  
BLWP @KSCAN perform keyboard scan  
LIMI 2 enable v d p interrupt so  
LIMI 0 you can press funct quit  
MOV @KEYVAL,R1 left byte R1 =key pressed  
MOV @STATUS,R0 left byte R0 =status byte  
RT return form BL @CALLKE  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\* defines colours,character  
\* DRAWING SUBROUTINE \* definitions, and puts the  
\*\*\*\*\* chars on the screen.  
DRAWBS LI R0,>390 colour table for char 128  
LI R2,3 only 3 colours to write  
BLWP @VMBW write colour bytes to vdp  
AI R1,4 adjust R4 to point to char  
MOV \*R1+,R3 get number of VMBW's to do  
MORDEF MOV \*R1+,R0 get address in vdp ram  
MOV \*R1+,R2 # bytes to move to vdp ram  
LOOP2 BLWP @VMBW define char or put on scre  
A R2,R1 adjust R1 to start mor vmb  
DEC R3 next data for vmbw statemt  
JNE MORDEF finished VMBWing?  
RT return to 1 of 5 routines  
\*\*\*\*\*

\*\*\*\*\*  
\* 5 ROUTINES FOR DRAWING PICTURES \*  
\*\*\*\*\*  
\*\* DRAW APPLE  
APPLE MOV R11,R10 save return address in R10  
LI R9,TAPPLE R9=addr of text for wordf  
LI R1,GAPPLE R1=addr of graphics data  
BL @DRAWBS BL (gосub) drawing routine  
B \*R10 return to calling program  
\*\*\*\*\*  
\*\* DRAW DOOR  
DOOR MOV R11,R10 save return address in R10  
LI R9,TDOOR R9=addr of text for wordf  
LI R1,GDOOR R1=addr of graphics data  
BL @DRAWBS BL (gосub) drawing routine  
B \*R10 return to calling program  
\*\*\*\*\*

\*\*\*\*\*  
\*\* DRAW CAR  
CAR MOV R11,R10 save return address in R10  
LI R9,TCAR R9=addr of text for wordf  
LI R1,GCAR R1=addr of graphics data  
BL @DRAWBS BL (gосub) drawing routine  
B \*R10 return to calling program  
\*\*\*\*\*  
\*\* DRAW SHIP  
SHIP MOV R11,R10 save return address in R10  
LI R9,TSHIP R9=addr of text for wordf  
LI R1,GSHIP R1=addr of graphics data  
BL @DRAWBS BL (gосub) drawing routine  
B \*R10 return to calling program  
\*\*\*\*\*  
\*\* DRAW HOUSE  
HOUSE MOV R11,R10 save return address in R10  
LI R9,THOUSE R9=addr of text for wordf  
LI R1,GHOUSE R1=addr of graphics data  
BL @DRAWBS BL (gосub) drawing routine  
B \*R10 return to calling program  
\*\*\*\*\*

\*\*\*\*\*  
\* SPELLS SECOND FILE \*  
\*\*\*\*\*  
SUBSV2 DATA 0 save return addr from subrsf  
BSS 32 workspace for subroutines  
\*\*\*\*\*  
\* REALLY LONG DELAY \*  
\*\*\*\*\*

\*\*\*\*\*  
DELA1 DATA SUBWS,DELA1 workspace,jump toDELA1  
DELA1 LI R7,1000 this subroutine can be  
DELA1 LI R8,250 called with BLWP @DELA1  
DELA1 DEC R8 at any time to slow down  
JNE DELA1 the program to debug it.  
DEC R7  
JNE DELA1  
RTWP  
\*\*\*\*\*

\*\*\*\*\*  
\* CLEARS \* clear screen routine  
\*\*\*\*\*  
CLEARS MOV R11,@SUBSV2  
BL @HCHARD  
DATA 0,SPACE,768  
MOV @SUBSV2,R11  
RT  
\*\*\*\*\* called by BL @RANDOM  
\*\*\*\*\*

\*\*\*\*\*  
\* RANDOMIZER \*  
\*\*\*\*\* returns random word in R1(0->FFFF)  
RANDOM LI R0,28645  
MPY @8300,R0  
AI R1,31417  
MOV R1,@8300  
MOV @8300,R1  
RT return to calling subtr  
\*\*\*\*\*

\*\*\*\*\*  
\* VCHAR \* write chars in a vertiacal  
\*\*\*\*\* pattern.  
VCHAR LI R3,1 vchar using registers  
JMP VCHARZ  
VCHARD LI R3,1 vchar by passing data  
JMP VCHARB  
\*\*\*\*\*

\*\*\*\*\*  
VCHARM MOV \*R11+,R3 multiple vchar's  
VCHARB MOV \*R11+,R0 move addr from data to R0  
MOV \*R11+,R1 move char to left byte R1  
MOV \*R11+,R2 move # bytes from data R2  
VCHARZ BLWP @VSW write first character  
DEC R2 last character?  
JEQ VCHARE if last one check for more  
AI R0,32 next row  
CI R0,>300 if not off screen  
JLT VCHARZ continue adding 32 to addr  
AI R0,-767 else adjust to start of  
JMP VCHARZ next row  
VCHARE DEC R3 last vchar ?  
JNE VCHARB if not go VCHARB (do more)  
RT return from subroutine  
\*\*\*\*\*

\*\*\*\*\*  
\* HCHAR \* write consecutive chars on  
\*\*\*\*\* screen  
HCHAR LI R3,1 hchar by passing values in  
JMP HCHARZ registers R0,R1,R2  
HCHARD LI R3,1 hchar by passing values in  
JMP HCHARB data statements.  
HCHARM MOV \*R11+,R3 R3= number hchars  
HCHARB MOV \*R11+,R0 R0= address on screen  
MOV \*R11+,R1 R1(MSB)=character to print  
MOV \*R11+,R2 R2=number of bytes  
HCHARZ BLWP @VSW write char on screen  
INC R0 next screen position  
DEC R2 last char ?  
JNE HCHARZ  
DEC R3 last hchar ?  
JNE HCHARB  
RT return  
\*\*\*\*\*

\*\*\*\*\*  
\* INTEGER TO FLOATING POINT TO ASCII STRING \*  
\*\*\*\*\*  
INTSIR BLWP @MLINK change integer to floating  
DATA >2300 point number in >834A  
FP2STR CLR @837C clear status byte  
CLR @8355 basic format for string  
BLWP @GPLINK convert f.p # to string  
DATA >14  
LI R1,>83 must add >8300 to addr in  
MOV @9355,R1 >8355!!!!!!!!!!!!!!!!!!!!  
SWPC S1!!!!!!!!!!!!S1=aeesesslog stsiog  
MOV @9356-R3  
!!!!!!SRM R3,9 R3=length of!suring  
CLS @837C cmeasStatus/indicate no err  
RT return  
\*\*\*\*\*

\*\*\*\*\*  
\* DATA FOR DEFINING COLOURS & CHARACTERS \*  
\*\*\*\*\*  
\*\* DATA TO DRAW APPLE, COLOUR, DEFINE, PRINT IT  
GAPPLE BYTE >6E,>2E,>6E,>00 4 colour codes  
\*\*\*\*\*



```

DATA 6      number of vmbw's to perform
DATA >0000,56 *0000=addr to define ch128
DATA >0000,>0000,>0000,>0E1F def char 128
DATA >0000,>0000,>0000,>70F8 def char 129
DATA >3F7F,>FFFF,>FFFF,>3F7F def char 130
DATA >FCFE,>FFFF,>FFFF,>FEFE char 131
DATA >3F3F,>1F1F,>0F0F,>0703 char 132
DATA >FFFF,>FFFF,>FFFF,>FF81 char 133
DATA >FCFC,>F8F8,>F0F0,>E0C0 char 134
DATA >0C80,8 *0C80=addr to defineCH 144
DATA >FFFF,>F3F3,>FFFF,>FFFF char 144
DATA >0C40,8
DATA >0020,>7018,>000C,>000C char 136
DATA >153,4 row 10
BYTE 128,136,129,32 to draw apple
DATA >173,4 row 11
BYTE 130,144,131,32 to draw apple
DATA >193,4 row 12
BYTE 132,133,134,32 to draw apple
EVEN if you make a mistake entering
** DATA TO DRAW DOOR, COLOUR, DEFINE, PRINT IT
GDOOR BYTE >4E,0,0,0 4 colour codes
DATA 4      number of vmbw's to perform
DATA >0000,40 *0000=addr to define ch128
DATA >FFFF,>FFFF,>FCFC,>FCFC def char 128
DATA >FFFF,>FFFF,>3F3F,>3F3F char 129
DATA >FCFC,>FCFC,>FFFF,>FFFF char 130
DATA >3F3F,>3333,>FFFF,>FFFF char 131
BLOCK DATA >FFFF,>FFFF,>FFFF,>FFFF char 132
DATA >153,2 row 10
BYTE 128,129 to draw door
DATA >173,2 row 11
BYTE 130,131 to draw door
DATA >193,2 row 12
BYTE 132,132 to draw door
EVEN
** DATA TO DRAW CAR, COLOUR, DEFINE, PRINT IT
GCAR BYTE >40,0,0,0 4 colour codes
DATA 3      number of vmbw's to perform
DATA >0000,64 *0000=addr to define ch128
DATA >0000,>0000,>0001,>0307 def char 128
DATA >0000,>0000,>FFC1,>C1C1 char 129
DATA >0000,>0000,>F008,>0402 char 130
DATA >0000,>0000,>0000,>0000 char 131
DATA >3F7F,>FFFF,>E66F,>0F06 char 132
DATA >FFFF,>FFFF,>7F7F,>0000 char 133
DATA >FFFF,>FFFF,>FEFE,>0000 char 134
DATA >F8FC,>FFFF,>67F6,>F060 char 135
DATA >153,4 row 10
BYTE 128,129,130,131 to draw car
DATA >173,4 row 11
BYTE 132,133,134,135 to draw car
EVEN
** DATA TO DRAW SHIP, COLOUR, DEFINE, PRINT IT
GSHIP BYTE >34,>E4,0,0 4 colour codes
DATA 4      number of vmbw's to perform
DATA >0000,32 *0000=addr to define ch128
DATA >0000,>0000,>0000,>0103 def char 128
DATA >3030,>3030,>30FF,>FFFF char 129
DATA >C0C0,>C0C0,>C0F0,>F8FC char 130
DATA >0000,>0000,>0000,>0000 char 131
DATA >0C40,32
DATA >FFFF,>FA7F,>3F1F,>0F0F char 136
DATA >FFFF,>BBFF,>FFFF,>FFFF char 137
DATA >FFFF,>BBFF,>FFFF,>FFFF char 138
DATA >FFFF,>BCF8,>F0E0,>C0E0 char 139
DATA >153,4 row 10
BYTE 128,129,130,131 to draw ship
DATA >173,4 row 11
BYTE 136,137,138,139 to draw ship
EVEN
** DATA TO DRAW HOUSE, COLOUR, DEFINE, PRINT IT
GHOUSE BYTE >64,>24,>A4,0 4 colour codes
DATA 6      number of vmbw's to perform
DATA >0000,24 *0000=addr to define ch128
DATA >0000,>0000,>0000,>0000 def char 128
DATA >0000,>0000,>0000,>0020 char 129
DATA >0000,>0000,>0000,>0000 char 130
DATA >0C40,24
DATA >0307,>0F1F,>3F7F,>BF3F char 136
DATA >FFFF,>FFFF,>FFFF,>FFFF char 137
DATA >C0E0,>F0F8,>FCFE,>FD1C char 138
DATA >0C80,24
DATA >3C3C,>3F3F,>3939,>3F3F char 144
DATA >E7E7,>FFFF,>FFFE,>E7E7 char 145
DATA >3C3C,>FCFC,>9C9C,>FCFC char 146

```

```

DATA >153,4
BYTE 128,129,130,32
DATA >173,4
BYTE 136,137,138,32
DATA >193,4
BYTE 144,145,146,32
EVEN
END

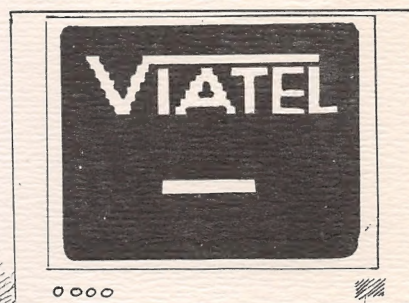
```

```

row 10
to draw house
row 11
to draw house
row 12
to draw house

```

**END**

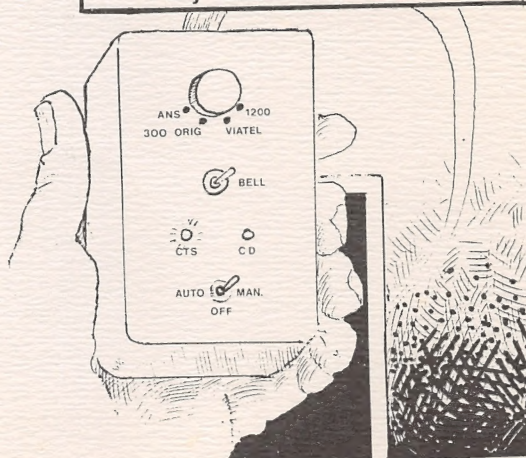


TEXAS INSTRUMENTS

## VIATEL MODEM CARD

TERRIFIC VALUE FOR MONEY.  
MADE BY TISHUG MEMBER,  
PHONE PETER NOW ON  
02 358 5602 FOR DETAILS  
AND AVAILABILITY.

300 & 1200 BAUD  
Originate or Auto-Answer  
Auto-Dial thru cassette port  
Fully Guaranteed





28